RHEL 5 » Ян Мердок » GNOME 2.18

LXF DVD

Elive Linux

ПЛЮС: Мегапак альтернативных ОС!

» 50+ страниц HOWTO
» VirtualBox » ClamAV

LINUX FORMAT

Главное в мире linux

Май 2007 № 5 (92)

Суперкомпьютер с Linux

2007 – год четырехядерных процессоров!

» Выбор комплектующих
» Сборка: пошаговое руководство
» Тесты » Советы

Red Hat

Готовитесь покорять новые вершины с RHEL 5? Прочтите наш обзор на с. 10

Учебник на 46 страницах »

4 607130 820513

Уничтожайте!

Лучшее оружие для войны со спамом с. 28

Творите!

Rosegarden и его новые мощные инструменты с. 13

Разрабатывайте!

Мобильные приложения для телефонов с Qtopia с. 50

«По-моему, Debian упускает ТАКУЮ возможность...»

Прямолинейный Ян Мердок с. 34

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882
Каталог «ПРЕССА РОССИИ» – подписной индекс 87874

Linux center
www.linuxcenter.ru

LINUX FORMAT
Главное в мире Linux

К Вашим услугам...

С собранным в этом месяце суперкомпьютером играл только Пол, и мы все ему завидуем. А потому спросили у команды LXF: Каково, по-вашему, наилучшее применение для четырехядерного процессора?

Пол Хадсон
Запуск четырех копий Unreal Tournament 2004. Ну, понеслась!

Грэм Моррисон
Отопление жилища зимой и пробуждение ежиков от спячки.

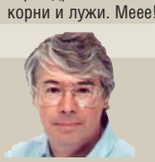
Майк Сондерс
Заполнение моего счета в баре Cattle Market Inn. Да я б и от Altix'а не отказался...

Эфраин Эрнандес-Мендоса
Я бы использовал его как подставку для кофе – достаточно массивно и солидно.

Ребекка Смолли
Для каких-нибудь сверхважных супервычислений, но чтобы оно непременно прыгало через древесные корни и лужи. Мееее!

Эндрью Грегори
Используйте одно ядро, как обычно, а три отдайте проекту SETI. Истина где-то там!

Энди Хадсон
Управление моей поливальной машиной. Мощности всегда не хватает!



Д-р Крис Браун
Одно ядро для вычислений, три – для подогрева кофе.

Нейл Ботвик
Пока не знаю, потому что у меня его нет. Пошлите мне парочку, и я обязательно отвечу на ваш вопрос.

Энди Ченнел
Решить проблему всемирного потепления – и поучаствовать в нем своим тепловыделением.

Скотт Дуглас
Наилучшим применением будет засунуть его в компьютер – не думаю, что он заработает в тостере.

Миссия журнала

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО

Linux – это что?

Не так давно мне довелось читать дискуссию, некоторым образом затрагивающую компанию Red Hat. Это была бы еще одна ничем не примечательная словесная баталия на поле «войн дистрибутивов», но в ее ходе был сформулирован интересный вопрос: «А что же такое стандартный Linux?» Мы гордимся присущей Open Source свободой выбора, возможностью «перекроить» любое приложение под собственные нужды, но при этом не всегда выдерживаем грань между разнообразием и разобщенностью. Red Hat делает для открытого ПО больше, чем кто-либо другой и (да простят меня наши друзья из Novell) фактически является стандартом на предприятиях. Существующие стандарты, вроде LSB, которые тоже во многом «списаны» именно с нее. Так что, Red Hat – это Linux? Я уже слышу недовольный ропот сторонников Debian, Gentoo, Slackware. «Мы ближе к Unix,» – доносится из их рядов.

В науке говорят, что некоторое явление по-настоящему понято исследователем лишь тогда, когда он способен объяснить его суть своей бабушке. А как в двух словах обрисовать Linux человеку, который последние 10 лет провел в застенках Редмонда? Подумайте и напишите нам об этом. **LXF**

Валентин Синицын » Главный редактор info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru

› **Адрес редакции:** Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54.
› **Телефон редакции:** (812) 717-00-37. Дополнительная информация на стр.118

LINUX FORMAT

Май 2007 № 5 (92)

Содержание

Весь номер – прямо как на ладони: приятного чтения!

Учебники

Система

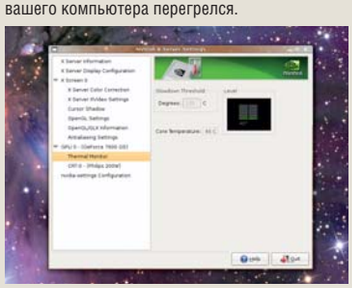
Резервирование на «раз-два» 54

Вогоны грозятся стереть ваш дом с лица Земли? Не беспокойтесь – с этим руководством вы успеете сохранить свои данные.

Оборудование

Мониторинг системы 58

Узнайте, как установить датчик, оповещающий вас о том, что в процессе нашествия процессор вашего компьютера перегрелся.



› **Сверхчувствительный контроль.**

SugarCRM

Собственный модуль 62

Заставьте CRM-систему соответствовать нуждам вашей организации на все сто – напишите все, чего не хватает, самостоятельно.

Mono

Объекты и обобщенные типы 66

Сядьте прямо, расправьте плечи, разомните пальцы – начинаем урок объектной ориентированности.

Ant

Улучшаем Java-проекты 70

В серии «Hardcore Linux»в этом месяце: заставить Ant работать и разобраться со всеми его задачами.

GTK

Первое приложение GNOME 74

Хватить ныть, что второй по популярности рабочий стол не создан для жизни – пора взять вожжи в свои руки и сделать все так, как вам нравится!

Unix API

Мышь и цвет 78

Старая добрая консоль едва ли способна работать в True Color, но установить на ней любимую палитру П. Нортона вполне реально.

Java EE

Паттерн MVC 82

Приготовьтесь слушать: сегодня мы познаем суть таинственного термина «паттерн».

LaTeX

Справочно-поисковый аппарат издания 86

Оглавления, сноски, ссылки – и все то, что отличает книгу от комикса.

Cinelerra

Хроматический ключ 91

Встречайте: динозавры на улицах вашего города!

LXF DVD92

Майк вам покажет 106

Elive

Мощь и стабильность Debian плюс изящество Enlightenment – самый красивый в мире оконный менеджер на расстоянии перезагрузки!

Мегапак альтернативных ОС

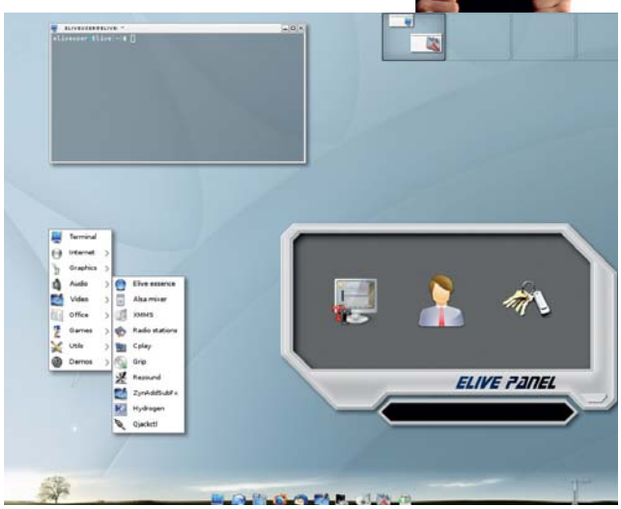
Open Source – это не только Linux: попробуйте нашу эксклюзивную подборку альтернативных ОС, включающую FreeBSD и Syllable.

Журнал в PDF

Более 50 страниц из предыдущих выпусков Linux Format: редактирование видео, управление пакетами, безопасный Firefox, настройка принтера и многое другое....

Новые релизы

Комплект обновлений и просто интересное ПО: KOffice 1.6.2, VirtualBox 1.3.6 и GPodder 0.9.0.



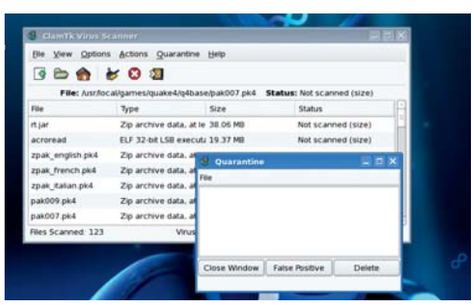
› ***Enlightenment*** **without the compilation hassles. Get it today!**

Что за штука... libferris?

Может ли RPM быть файловой системой? Почему бы и нет! с. 38

LXFHotPicks

Лучшие новинки открытого ПО на планете..... 100



› **Новинка этого сезона: элегантный интерфейс к** ***ClamAV*****.**

Отсеки спаму голову с. 28

Лучшее, что может предложить *SpamAssassin*

Подпишись на Linux Format и сэкономь!

Содержание

LXF DVD внутри!

См. страницу 106

Спецрепортаж

Суперкомпьютер с Linux

Подберем самые мощные комплектующие и заставим их работать на полную катушку! с. 40

А также...

Спаму.net 28
Наше терпение лопнуло – объявляем войну и набираем рекрутов!

Разбираемся с файлами журналов 30
Учимся выуживать из этого моря информации ценные сведения.

Программируем Greenphone 50
Разработка встраиваемого ПО: телефон не требуется

Школа LXF 114
Open Source и методички.

Постоянные рубрики

Новости 04

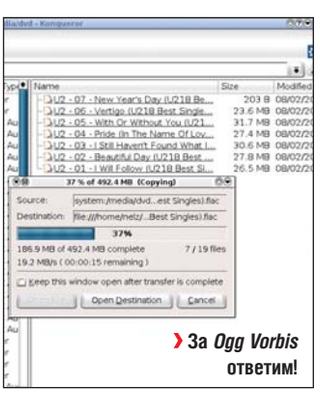
DistroWatch 26
Поостыв от разборок с Gentoo, Ладислав Боднар увлекся Linspire, а также нашел время взглянуть на Etch и SUSE 10.3..

Интервью LXF 34
Ян Мердок, основатель Debian, делится с Linux Format своими взглядами на Ubuntu, стандарты и...

Что за штука... 38
Автор libferris разговорился сам с собой, обсуждая, что же он создал. Это немного смущает...

Ответы 94
В этом месяце: Grub и RAID, fsck, Wine, планшеты и неловкий инсталлятор Fedora. Следующий!

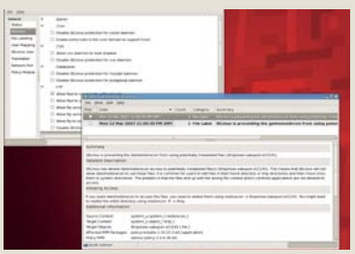
Через месяц 118
Что нам готовит LXF93.



› За *Ogg Vorbis* ответим!

Обзоры

RHEL 5 10
SUSE, Oracle и Ubuntu обзавелись серверными версиями – что может противопоставить им Red Hat?



› **RHEL 5: enterprise-ready, solid and stable – and now with AIGLX!**

Thunderbird 2 12
Что может предложить новинка от Mozilla матерым пользователям KMail и Evolution?

Rosegarden 1.5 13
...и почему его поддержка программных синтезаторов лучше, чем в Rosegarden 4.0.

Gnome 2.18 14
Новый Control Center, опция Gnibbles и утилита для создания снимков экрана. Да, нас тоже не завлекает.

Genius Look 316 15
В web-камерах обычно используется один из четырех чипсетов, и все они должны работать в Linux, верно?

Komodo IDE 4.0.2 16
Прекрасный подарок для программиста-полиглота.

Blender 2.43 18
Что разработчики умудрились упихать в очередное 0,4-процентное увеличение номера версии?

Сравнение: системы управления версиями

Bluefish 21
Seamonkey Composer 21
Nvu 22
Quanta Plus 23
Quanta Gold 23
Screem 24
OOo Web 24

ГЛАВНЫЕ НОВОСТИ: » Debilan » РИФ-2007 » Compiz и Beryl – трехмерная интерференция » Pidgin » GPLv3 выходит на финишную прямую » Mandriva Flash 4G » Пальмовый пингвин

» Рубрику ведет Илья Шпаньков

Кризис среднего возраста

В последнее время проект Debian переживает не лучшие дни. Многие пользователи отмечают некоторый застой в работе команды, проявляющийся в медленном исправлении ошибок, постоянных задержках с выпуском новых версий дистрибутива и многом другом. В конце марта сообщество разработчиков и пользователей этой самого свободной версии GNU/Linux повергла в уныние и еще одна новость: основатель Debian Ян Мердок [Ian Murdock] окончательно покинул проект, заняв пост директора отдела стратегического развития настольных операционных систем в компании Sun Microsystems. В персональном блоге он объяснил свое решение давнишним увлечением операционной системой Solaris и желанием поучаствовать в ее создании, но более реальные причины были высказаны им в интервью журналу *Linux Format* (см. стр. 34). По словам Мердока, проект находится в кризисе и основной причиной этого положения дел является слишком сильное увлечение разработчиков дистрибутива демократией. Коллегиальное обсуждение возникающих проблем приводит к значительному замедлению процесса их ликвидации, при этом совершенно отсутствует лидер, который мог бы волевым решением направлять процесс разработки в определенное русло.

Впрочем, не так все грустно, как выглядит на первый взгляд. По крайней мере, два события, последовавших за уходом Мердока и практически совпавших по времени, вселяют достаточную долю оптимизма. Во-первых, 9 апреля наконец объявлен долгожданный релиз операционной системы Debian GNU/Linux 4.0 (Etch). После двух лет, прошедших с появления прежней версии под номером 3.1 разработчиками была проделана огромная работа, позволившая сразу перейти к «четверке» в нумерации. В общей сложности было обновлено более 18000 пакетов, при этом вопреки устоявшемуся мнению о создателях Debian, как о консерваторах, предпочитающих слегка устаревший, но более стабильный код, в новую версию дистрибутива вошли практически самые свежие на тот момент компоненты – ядро Linux 2.6.18, графический сервер *X.Org 7.1*, настольные среды KDE 3.5 и GNOME 2.14, пакет офисных приложений *OpenOffice.org 2.0.4a*, графический редактор *GIMP 2.2.13*, HTTP-сервер *Apache 2.2.3* и многое другое. Также пользователи новой версии Debian получили обновленный графический инсталлятор, возможность работы с зашифрованными разделами, а для владельцев предыдущего релиза 3.1 разработчики предусмотрели механизм автоматического обновления операционной системы.

Второе важное для сообщества Debian событие – завершение ежегодной процедуры выборов нового лидера проекта. Им стал Самуэль Хосевар [Samuel Hocevar] из Франции, являющийся активным разработчиком данного дистрибутива с 2000 года. В своей предвыборной программе он наметил несколько важных направлений развития проекта, призванных слегка стряхнуть налет консерватизма и вновь сделать дистрибутив привлекательным (как истинный француз, Самуэль употребляет термин «сексуальным») для широких масс пользователей. При этом новый руководитель проекта считает, что участникам разработки вполне по силам сохранить массивность Debian, но одновременно сделать его динамично развивающимся проектом и, как следствие, вновь занять лидирующие позиции в мире свободного ПО. Сможет ли Самуэль реализовать намеченное – покажет время, но результаты голосования говорят о том, что именно такой вариант наиболее востребован сегодня у приверженцев самого свободного дистрибутива GNU/Linux.

› Самуэль Хосевар

РИФ-2007: свободные решения

В начале апреля прошло, пожалуй, наиболее яркое событие в жизни российского интернет-сообщества – 11-й Российский интернет-форум РИФ-2007. Надо отметить, что в этом году мероприятие оказалось еще более посещаемым: около 2200 человек приехали обсудить текущие вопросы развития интернет-отрасли, причем, помимо уже традиционных гостей из ближнего зарубежья, в форуме приняли участие и представители таких стран, как США, Австрия, Англия и Нидерланды. И такой интерес не случаен: по данным Роспечати к 2006 году отечественная интернет-аудитория выросла до 26 миллионов, а в 2007 году ожидается увеличение этой цифры еще на три миллиона пользователей, при этом даже представители правительства и президент России активно используют в последнее время всемирную сеть в качестве инструмента для ведения прямого диалога непосредственно с жителями страны. Поэтому и результаты обсуждений, прошедших в рамках данного мероприятия, имеют вполне определенный вес и могут повлиять на дальнейшее развитие интернет-отрасли в целом.

Среди прочих обсуждений (а на конференции прозвучало около 250 докладов) особо нужно отметить прошедший в рамках РИФ-2007 круглый стол на тему «Свободное ПО против коммерческого: риски и перспективы для России». Участники обсуждения, являющиеся представителями сообщества Open Source, бизнесменами и известными интернет-деятелями, обрисовали ряд важных проблем, существующих на сегодняшний день в связи со свободным программным обеспечением. Прежде всего, это отсутствие в российском законодательстве правовых норм, способных регулировать разработку и использование свободного ПО. Это приводит к полной незащищенности приверженцев Free Software как от разработчиков проприетарного ПО, так и со стороны правоохранительных органов. В связи с этим участники круглого стола приняли резолюцию, в которой поставили в разряд первоочередных задач необходимость тесного взаимодействия с законодательными органами в целях скорейшего придания свободному ПО юридического статуса, а также официального принятия стандарта ISO на открытый формат документов Open Document Format. Эти действия позволят значительно сократить расходы государственных организаций и частных предприятий на создание современных IT-инфраструктур за счет использования решений на базе свободного ПО.

Фото Макса Борзова. www.rif.ru

» Читайте репортаж с Круглого стола на стр. 7

http://linuxcenter.ru/lib/press/rif_2007_full.phtml

Трехмерная интерференция

Пример, достойный подражания, явили сообществу два проекта, работающие над созданием композитных рабочих столов, имитирующих трехмерность. После нескольких недель обсуждения, разработчики *Compiz* и *Beryl* решили объединить усилия, дабы не тратить время на решение, в общем-то, одних и тех же проблем. В результате слияния проект *Compiz* разделится на две ветки: *Compiz*-core будет содержать стабильный проверенный код и продолжит развитие силами прежних участников данного проекта, а *Compiz*-extra возьмет на себя роль тестовой лаборатории, в которой будут проходить обкатку все новшества и спорные идеи. При этом коллектив *Beryl* всем составом переходит именно в тестовую ветку проекта, получающую также временное название “Composite Community” и полностью контролируемую сообществом разработчиков свободного ПО (планируется открыть для этого специализированный web-сайт). После тщательного тестирования новые расширения и модули композитного рабочего стола будут постепенно передаваться в стабильную ветку для последующего использования в дистрибутивах GNU/Linux.

Следует напомнить, что в сентябре 2006 года проект *Beryl* образовался в качестве независимой ветки *Compiz*, разработка которого контролировалась компанией Novell. В отличие от традиционной для крупных корпораций немного осторожной политики разработки, участники *Beryl* ставили своей целью максимально быстрое внедрение самых новых идей в области трехмерных рабочих столов и более тесное сотрудничество с сообществом. При новом распределении сил команда *Beryl*, не нарушая собственных убеждений, будет попрежнему работать на самом авангардном направлении, а Linux-пользователи окажутся в выигрыше за счет освобождения от необходимости мучительного выбора между двумя практически идентичными решениями.

Что в имени тебе моем...

Популярная программа для обмена мгновенными сообщениями Gaim поменяла свое название на Pidgin и этому предшествовал многолетний конфликт с американским медиагигантом America On Line (AOL). Дело в том, что изначально программа называлась *«GTK+* AOL Instant Messenger» и, естественно, самой AOL это не понравилось. Тогда автор проекта Марк Спенсер [Mark Spencer] сократил название до аббревиатуры GAIM, что на тот момент вполне устроило американцев. Но через несколько лет медиакорпорация зарегистрировала название собственного клиента для обмена сообщениями – AIM, в связи с чем претензии к проекту Спенсера возобновились. При этом надо отметить, что споры вокруг имени шли с переменной активностью: одно время AOL перестала докучать свободному проекту и внешне казалось, что теперь все претензии остались в прошлом. Но накануне выхода Gaim 2.0 представители американского гиганта в почти ультимативной форме запретили пользоваться данным названием, что послужило причиной задержки релиза и искусственного продления стадии бета-тестирования. В конце концов, Спенсеру это надоело и он решил просто поменять название на Pidgin. Одновременно были переименованы библиотека «libgaim» (теперь «libpurple») и консольный вариант клиента «gaim-text» (теперь «Finch»). Такие шаги должны раз и навсегда избавить разработчиков Pidgin от «именных проблем», но данная история заставляет задуматься над тем, почему коммерческие компании обладают более высоким приоритетом перед свободным ПО. Можно с уверенностью сказать, что будь юридический статус свободного программного обеспечения более весомым, даже крупным корпорациям не пришло бы в голову заставлять менять название свободного проекта в связи с тем, что оно им тоже нравится.

www.pidgin.im

2 хорошо, а 4 – лучше

Разработчики компании Mandriva предполагали, что выпущенный в декабре прошлого года дистрибутив Mandriva Flash, предустановленный на флэш-брелок емкостью 2 гигабайта, будет иметь определенную популярность, но реальный спрос превысил все самые радужные прогнозы: идея иметь в кармане готовую операционную систему Mandriva 2007 (к тому же – с трехмерным рабочим столом), способную загружаться практически на любом компьютере, оказалась мега-популярной. Окрыленные успехом, разработчики не стали откладывать в долгий ящик удачное решение и к весне подготовили новый вариант – Mandriva Flash 4G. Помимо увеличения свободного места на флэш-брелоке до 3 ГБ (1 ГБ занимает собственно операционная система), пользователи получают дистрибутив Mandriva 2007, укомплектованный всеми свежими обновлениями. В состав операционной системы вошли графическая среда KDE 3.5.4, браузер *Firefox 1.5.0.10*, пакет офисных приложений *OpenOffice.org 2.0.4*, графический редактор *GIMP 2.3.10*, медиаплейер *RealPlayer 10.0.8.805*, программа для воспроизведения флэш-анимации *FlashPlayer 9.0.31.0* и многое другое. Стоимость четырехгигабитного варианта Mandriva Flash составляет 89 евро. Будет ли новинка продаваться в России, пока не ясно.

GPLv3 выходит на финишную прямую

Вынужденная задержка в обсуждении чернового варианта третьей версии лицензии GPL оказалась сравнительно недолгой – в конце марта опубликована очередная подкорректированная редакция, которая на этот раз была более благосклонно воспринята прежними критиками, к числу которых относился и Линус Торвальдс [Linus Torvalds]. Разработчики постарались сделать текст максимально приближенным к традиционным юридическим документам, при этом многие разделы стали проще и понятней, а некоторые «туманные» пункты были вообще удалены. Также отмечено стремление авторов сделать лицензию более универсальной, что позволит легче использовать ее в условиях индивидуальных особенностей законодательства различных стран. Также изменения коснулись и тех самых спорных моментов, связанных с техническими средствами защиты авторских прав (DRM) и TiVo-низацией. По мнению многих наблюдателей, формулировки стали значительно мягче, что позволило ярым критикам жестких мер отнестись к GPLv3 более заинтересованно. Линус Торвальдс даже не исключил, что при таком виде текста лицензии он может подумать над возможностью перевода ядра Linux на использование третьей версии GPL. Что касается Novell, чей договор с Microsoft и стал основной причиной корректировки графика выхода финальной версии текста лицензии, то на данный момент юристы компании не видят в тексте лицензии каких-либо пунктов, способных помешать выпуску Linux-дистрибутивов компанией Novell и в будущем.

Одновременно с возобновлением активного обсуждения новой редакции, представители Free Software Foundation озвучили и новое расписание подготовки конечного варианта. С момента выхода мартовского черновика (28 марта) в течении 60 дней будет вестись обсуждение и корректировка текста, а еще через 30 дней (конец июня) должен появиться долгожданный третий вариант лицензии GPL. Примечательно, что эта дата может стать началом нешуточной борьбы за сферы влияния на все свободное ПО в целом: компания Sun

Microsystems планирует максимально использовать разногласия в стане разработчиков GNU/Linux в отношении GPLv3, чтобы попробовать перетянуть часть сообщества на Solaris (как это произошло с основателем Debian), поэтому то, насколько внимательно отнесутся авторы текста GPLv3 к замечаниям со стороны, будет иметь вдвойне важное значение для сохранения традиционной сплоченности многомиллионной аудитории пользователей и разработчиков свободного ПО.

http://gplv3.fsf.org/gpl-draft-2007-03-28.html

IBM

ПРОБЛЕМ МЕНЬШЕ. ПРОИЗВОДИТЕЛЬНОСТЬ ВЫШЕ.

Представляем платформу System p, лидирующую среди более чем 70 ведущих систем[1]. Она не только обладает всеми преимуществами передовых технологий, но и позволяет избежать обычных проблем, связанных с недостаточной производительностью. Платформа обеспечивает беспрецедентную гибкость в управлении растущими базами данных и расширяет возможности контроля над взаимодействием серверов и затратами на их обслуживание. Кроме того, System p гарантирует превосходную производительность серверов.

Подробнее о том, как System p может помочь Вам достигнуть большего с меньшими трудностями, – на сайте **ibm.com**/systems/ru/p

[1]Серверы IBM System p5 и eServer p5 удерживают первое место среди 70 систем. Подробнее на http://ibm.com/systems/p/benchmarks. Представленные результаты действительны на 16.11.06. Источники: http://www.tpc.org, http://www.sap.com/benchmark. IBM, логотип IBM, System p и POWER являются товарными знаками или зарегистрированными товарными знаками International Business Machines Corporation в США и/или других странах. Наименования других компаний, продуктов и услуг могут быть товарными знаками или знаками обслуживания третьих лиц. © 2007 IBM Corporation. Все права защищены.

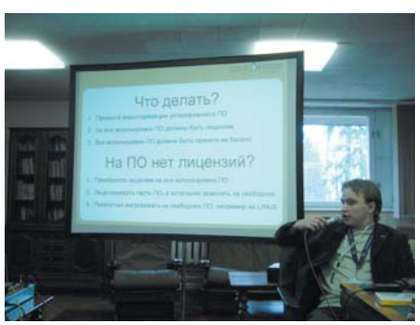
Реклама

Экстрановость

РИФ-2007: Круглый стол глазами участника

Своими впечатлениями от последнего события на российской сцене свободного ПО с читателями *LXF* делится **Иван Зорин**.

Расписание мероприятий было сдвинуто с самого утра, именно поэтому, вместо положенных четырнадцати-ноль-ноль, круглый стол «Свободное ПО против коммерческого: риски и перспективы для России» был начат приблизительно в полтретьего. К сожалению, те презентации и доклады, которые должны были быть показаны и рассказаны, не состоялись или состоялись не в том виде. Именно в силу ограниченности временного регламента участников постоянно торопили. Некоторых также опечалило то, что Круглый стол не смог посетить Поносов Александр Михайлович, а телемост с ним не получилось наладить «по техническим причинам». Несмотря на то, что Павлу Фролову удалось продемонстрировать некоторые слайды презентации и изложить свой взгляд на положение свободного ПО в России, а Павлу Житнюку – отметить некоторые моменты, в конечном итоге получился круглый стол в самом прямом смысле этого слова. Была поднята актуальная тема об использовании свободного ПО в секторе частного бизнеса, где проверок на «лицензионную чистоту организации» практически не избежать. Ситуацию призваны спасти переговоры, которые активно пытаются вести представители Линуксцентра с мировыми компаниями-разработчиками и дистрибьютерами свободного ПО. Так, на сегодняшний день налажены контакты с компаниями Canonical Ltd (производитель Ubuntu) и с Mandriva, которые готовы предоставлять «бумажные» лицензии на собственное свободное ПО. В проекте разработка коробочной версии Ubuntu, включающей в себя в печатном виде лицензию с основными положениями на свободное использование. Однако, применение дистрибутивов Linux, разрабатываемых децентрализованно программистами-энтузиастами со всего света, когда не существует официального представительства компании-разработчика (Gentoo, Debian и им подобные) на предприятии в сегодняшних российских реалиях подобно бегу по лезвию ножа.



Ничуть не удивило появление на подобном мероприятии представителя корпорации Microsoft – Петра Диденко. Не без его участия обсуждался вопрос о нецелесообразности использования проприетарного ПО в военно-промышленном комплексе нашей страны. Г-н Диденко даже выступил с оригинальным предложением: «Поднимите руку те, кто хотя бы раз открывал исходник какой-либо программы». Ответом был лес рук присутствующих и неудачность вопроса была признана самим автором.

На Круглом столе был анонсирован набор свободного ПО для Windows, занимающий 1 CD и включающий в себя аналоги коммерческих продуктов, такие как графический редактор GIMP, web-браузер Firefox, офисный пакет OpenOffice.org и другие подобные приложения, широко известные в Linux-среде.

Участникам стола также была предоставлена возможность опробовать Mandriva Flash – дистрибутив Mandriva Linux на 2 ГБ USB-накопителе.

По итогам круглого стола можно с уверенностью сказать, что в России действительно есть люди и компании, которым небезразлична судьба свободного ПО вообще и операционной системы Linux в частности, так что, возможно, массовая миграция конечных пользователей с проприетарных систем на свободные – всего лишь вопрос времени. Приятно, что существуют организации, которые своей деятельностью всячески стараются этому способствовать.

Пингвин на пальме

Наконец стали известны некоторые подробности о Linux-платформе, разрабатываемой компанией Palm для своих устройств линейки Treo. На прошедшей в Нью-Йорке встрече «Palm Investor and Analyst Day» глава компании Эд Коллиган [Ed Colligan] сообщил, что к концу года будет официально представлена новая платформа на базе Linux, полностью совместимая с Garnet OS (Palm OS 5). Среди основных и давно ожидаемых возможностей новая операционная система будет обладать многозадачностью и полноценной поддержкой Wi-Fi. В качестве ключевого интернет-приложения, в обязанности которого будет входить обеспечение полноценного отображения web-контента на небольшом экране устройств, предполагается использовать норвежский браузер Opera, обладающий оригинальными технологиями Small Screen Rendering (SSR) и Medium Screen Rendering (MSR). Также в компании Palm надеются, что использование Linux позволит привлечь большое число сторонних разработчиков к созданию прикладных программ.

Новости короткой строкой

» В Пермском техническом университете завершился процесс миграции с операционной системы Windows на Mandriva 2007.

» Очередное обвинение по статье 146 УК РФ за использование нелицензионного программного обеспечения предъявлено директору челябинского текстильного комбината Владимиру Калиниченко.

» Владимир Путин подписал федеральный закон «О внесении изменений в статьи 146 и 180 Уголовного кодекса Российской Федерации», согласно которому срок лишения свободы за использование нелицензионного ПО увеличен с 5 до 6 лет.

» Активисты нижегородской Linux Users Group открыли русскоязычный вариант проекта «Пакет дня Debian».

LINUX FORMAT

Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов

Алексей Федорчук
Свою первую (и последнюю) программу написал еще на Алголе.

Обновление Debian-семейства

Знаковые события уходящего месяца – превращение Debian Etch в стабильный релиз (за номером 4.0) и выход Ubuntu (во всех ее ипостасях) версии 7.04. Если второе случилось точно по графику, то первое – с некоторым запозданием (впрочем, в масштабах времени Debian'а квартальное опоздание можно считать лежащим в пределах ошибки эксперимента). И, тем не менее, сопряженность выхода этих двух дистрибутивов – символична. Два года назад, когда началось триумфальное шествие Ubuntu по пользовательским ПК, возникло не лишенное оснований опасение – а не приведет ли это к кончине ветерана дистростроения?

Ныне можно констатировать, что опасения оказались напрасными. Debian не только не умер – появление собрата-конкурента придало его развитию динамизм. Если раньше, сколько я помню себя в Linux'е, «тормознутость» в выпуске его версий была притчей во языцех, и трехлетний релиз-цикл воспринимался как должное, то ныне мы видим существенное сокращение сроков. И графический инсталлятор Debian, о необходимости которого так много говорили большевики (пардон, разработчики), наконец, увидел свет. Причем, в отличие от Ubuntu, в виде, функционально идентичном инсталлятору текстовому.

Будем надеяться, что «мирное сосуществование» отцов и детей в Debian-семействе будет продолжаться, к вящей пользе тех, кто применяет того или иного его представителя.

alv@posix.ru

Сегодня мы рассматриваем...

10 RHEL 5
Пока Novell закреплялась в серверном сегменте рынка, Red Hat готовила убойную промышленную ОС. Удалось ли ей это?

12 Thunderbird 2
Его так долго ждали, и он все-таки прилетел! Мы скачали последний релиз-кандидат и внимательно изучили его, чтобы понять, чего ожидать от новой версии

13 Rosegarden 1.5
Сначала был Rosegarden 2.0, затем – 4.0, а сейчас вот добрались и до 1.5. Если вы видите здесь какой-либо смысл, то заблуждаетесь, но путь к совершенству всегда тернист и извилист.

14 Gnome 2.18
Точнее атомных часов, очередное обновление Gnome снова здесь. Стоило ли ждать полгода, чтобы обзавестись утилитой для создания снимков экрана и парой других штучек?

15 Genius Look 316
Да, это обзор web-камеры. Нет, вы не ошиблись журналом. В этом месяце мы решили изменить традициям – иногда страсть к «железу» захватывает нас без остатка.

16 Komodo IDE 4.0
XUL + Mozilla Application Framework + больше инструментов, чем в швейцарском армейском ноже. Достаточно ли этого, чтобы получилось идеальная машина для кодирования?

Blender 2.43 с.18

18 Blender 2.43
В очередную «сотую» вместились режим скульптора и улучшенный симулятор жидкостей – что еще таится за столь скромным изменением номера версии?

Genius Look 316 с.15

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth
Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.*

Рейтинг 9/10

Red Hat Enterprise Linux 5

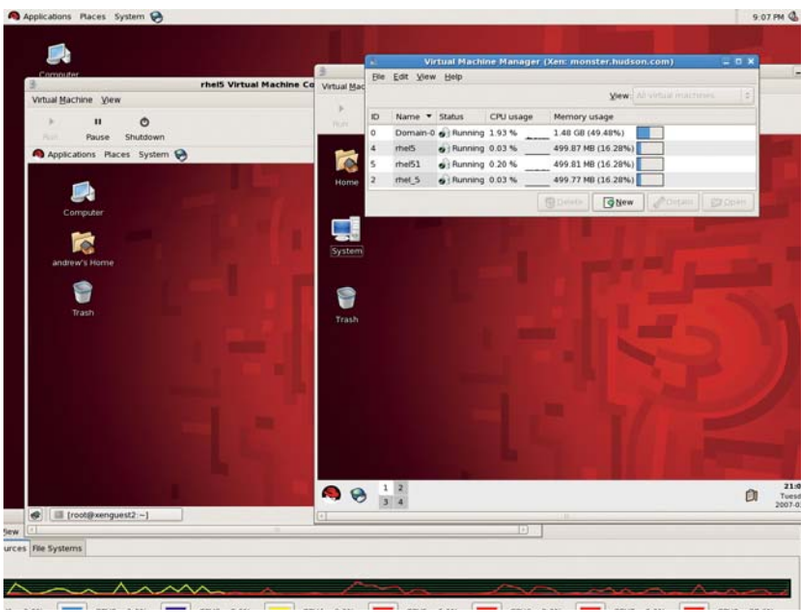
Red Hat вбросила новый продукт на рынок с ужесточающейся конкуренцией. **Энди Хадсон** смотрит, приняты ли меры, чтобы унять выскочек.

Вкратце...

» Linux-дистрибутив для предприятий. См. также: SLES, Unbreakable Linux от Oracle, CentOS.

Нечасто нам случалось рассматривать дистрибутивы настоящего бизнес-класса. Да и выбора не было: Red Hat Enterprise Linux просто не с чем было сравнивать. Но от этой, пятой, версии ожидают очень многого. Теперь у Red Hat есть сильный соперник не только в лице SUSE, но и Oracle с перекомпилированной версией самого RHEL! Достаточно ли сделала компания из Северной Каролины для того, чтобы остаться лидером?

Прежние версии Red Hat Enterprise Linux четко делились на четыре группы: ES (Enterprise Server), AS (Advanced Server), WS (Workstation) и Desktop, каждая с особым набором пакетов и областью применения. В RHEL все по-другому – здесь только два варианта: Server для серверов и Desktop для настольных компьютеров. Выбор упрощается; но это еще не все. Вместо класса инсталляции появился инсталляционный ключ, он генерирует готовый преднастроенный набор пакетов. У RHEL 5 Server таких ключей два: один предоставляет базовый сервер с ограничением до четырех виртуальных машин, а второй служит для установки 'Advanced Platform', с расширенными возможностями по виртуализации и памяти. Настольный вариант предлагает четыре предопределенных набора пакетов, которые оптимизированы для обычного ПК, технической рабочей станции, а также различных аспектов виртуализации.



› **Год назад Red Hat утверждала, что *Xen* не готов к бизнес-использованию – теперь технология виртуализации прошла тесты Red Hat и стала главной частью Enterprise Linux.**

> «От Up2date отказались в пользу Yum, Pirut и Pup.»

Старый змий

Мы тестировали RHEL 5 Server и Desktop на двухпроцессорном сервере Quad Core Intel Xeon (Clovertown). Установка необычайно проста, ибо RHEL пользуется проверенной утилитой инсталляции *Anaconda*, и сходна с установкой Fedora, «общественного» варианта Red Hat; основное отличие – новый этап ввода ключа для автоматической установки необходимого набора пакетов. Впоследствии набор можно изменить, но мы предпочли за каких-то 20 минут получить Advanced Platform. По умолчанию оба серверных варианта загружаются прямо в *Xen*, давая полную свободу виртуализации уже на старте.

Важные показатели

Red Hat Enterprise Linux базируется на ядре 2.6.18, расширенном для нужд предприятий, и раз уж это бизнес-дистрибутив, версия ядра меняться не будет: предусмотрена методика заплаток и обратного портирования из новых ядер. Вся система скомпилирована с использованием *GCC 4.1* и *Glibc 2.4*, и благодаря новым алгоритмам GCC повысилась производительность. RHEL 5 удвоила вместимость файловой системы ext3 до колоссальных 16 ТБ, хотя это только технологический аванс, с полной поддержкой будущих обновлений.

Усовершенствована интеграция *Samba* с Active Directory, есть улучшения и в SELinux. Для незнакомых с этим понятием, поясним: с помощью SELinux ядро изолирует друг от друга работающие процессы (например, буйные демоны *Apache* или *httpd*) и выдает им только сугубо необходимые привилегии. SELinux также пересмотрен, с введением *system-config-selinux* для глубинной настройки политики. Он также обзавелся Multi-Level Security (MLS), многоуровневой безопасностью, и теперь Red Hat формально может быть аккредитован по EAL4+/LSPP для работы в сверхсекретных правительственных и военных учреждениях США.

Давние пользователи Red Hat (как RHEL, так и первоначального Red Hat Linux) заметят отсутствие *Up2date*, старинного средства отслеживания и загрузки обновлений: вслед за Fedora, выбор сделан в пользу *Yum*, *Pirut* и *Pup*. Red Hat предпочла *Yum* ради его модульного принципа работы, позволяющего RHEL 5 использовать новый *rhn_plugin* для доступа к содержимому Red Hat Network, сети для администраторов. Включение *Yum* заодно обеспечивает доступ к хранилищу сторонних программ – но помните, что ПО, установленное из любого стороннего хранилища, не поддерживается ни при каком варианте приобретения RHEL. Другая причина принятия *Yum* – это мощная поддержка сообществом во время обкатки в Fedora (*Up2date* в основном варилась в собственном соку).

Red Hat на Xen

Linux Format обстрелял Брайана Стивенса [Brian Stevens], руководителя техотдела Red Hat, вопросами о важности *Xen* для RHEL 5.

Linux Format: Виртуализация – козырь RHEL 5. Как Red Hat удалось поднять *Xen* до уровня предприятия?

Брайан Стивенс: Это итог двух с половиной лет работы над стабильностью и производительностью. Применение виртуализации непосредственно в RHEL настолько повысит производительность и управляемость, что позволит потягаться с последователями *VMware*.

LXF: Эта технология всегда была непроста в использовании – какие инструменты управления доступны для *Xen*?

БС: Средства управления развиваются стремительно. На http://libvirt.org можно найти набор виртуализационных API, на базе которых программисты и поставщики ПО могут строить инструменты управления. http://virt-manager.et.redhat.com распространяет *libvirt*, обеспечивая виртуальным машинам (ВМ) графические инструменты управления созданием, миграцией, потреблением ресурсов. RHEL 5 включает систему мониторинга и перехвата отказов ВМ. Red Hat Network Satellite (сеть, используемая для доставки и обновления ПО) расширена, и теперь может доставлять и обновлять ВМ.

Теперь о рабочем столе. В состав RHEL 5 Desktop вошел долгожданный *OpenOffice.org 2.0.4* (в RHEL 4 была гораздо более старая версия). Значительно улучшены фильтры для форматов Microsoft, а также поддержка набирающего популярность Open Document Format (ODF). Включены Gnome 2.16 и *Compiz*, с добавкой популярных эффектов (например, вездесущих крутящихся кубиков). Больше всех повезло владельцам ноутбуков, так как RHEL 5 Desktop безупречно работает с такими функциями, как ACPI и подключение по беспроводной сети с помощью умелого *NetworkManager*.

Многоглавый

В прошлом году Red Hat публично принижала важность *Xen*, после того как Novell объявила о включении виртуализации в SLES 10. В сущности, поведение Red Hat просто отвечало природе стремительного развития *Xen*: ведь на той младенческой стадии отсутствовал даже реальный графический интерфейс настройки. Поэтому Red Hat твердо решила дождаться «повзросления» *Xen*. С тех пор прошло восемь месяцев, за которые команда успешно продвинула *Xen* в эпицентр дистрибутива. *Xen* стала изюминкой RHEL 5 как для серверного, так и для настольного вариантов; оба сервера включают ее по умолчанию, поэтому можно загружаться прямо в ядро с *Xen*-поддержкой. *Xen* доступна на платформах i386 и x86_64, с полной поддержкой Itanium после обновления Update 1 (ожидается в ближайшие месяцы). Отшлифованный *Virtual Machine Manager* после пробы в Fedora Core 6 с удовольствием принят в RHEL, с ним инсталляция и настройка виртуальных машин на *Xen*-основе стала относительно проще.

Пол считает...

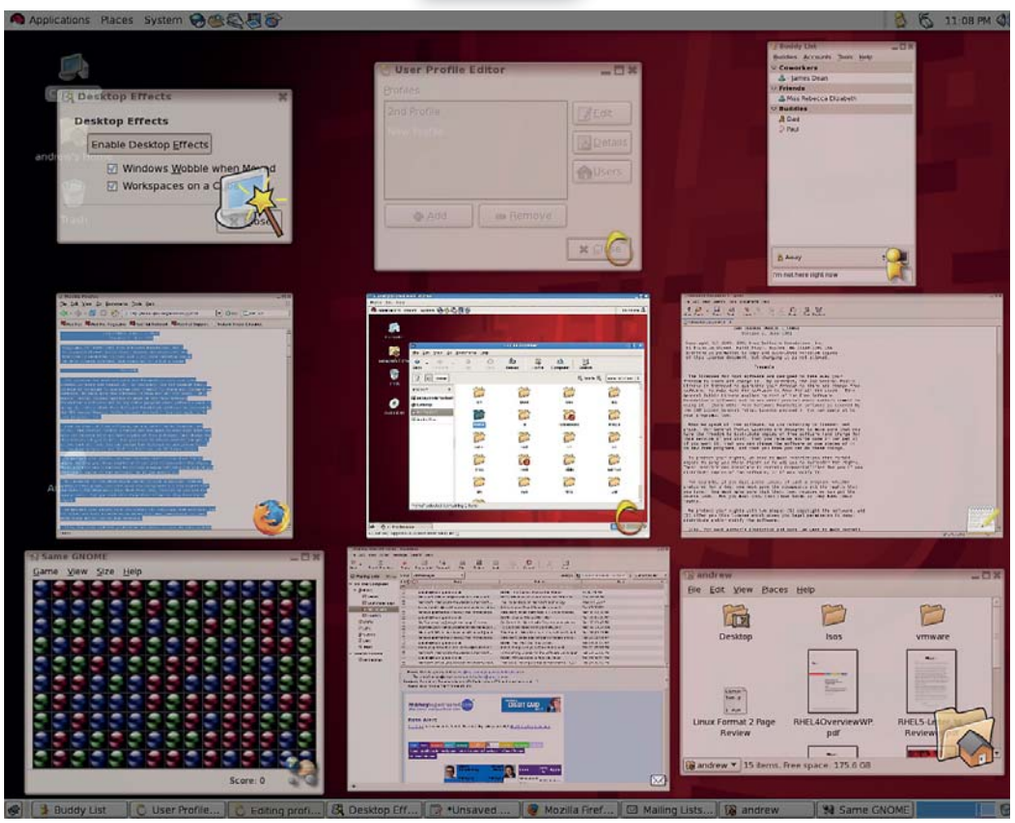
«Учтите: у RHEL очень специфические требования к серверу. Если ваша аппаратура не имеет сертификата Red Hat, попробуйте сначала демо-версию!»

Еще один технологический аванс – система Stateless Linux, позволяющая запускать и поддерживать несколько настольных RHEL из одного централизованного образа. Stateless Linux – совершенно новый метод реализации Linux на корпоративном уровне, и хотя в первоначальном выпуске RHEL 5 поддерживается не полностью, но определенно заслуживает внимания в среде разработчиков.

Так держать!

RHEL 5 станет важной вехой в истории Red Hat. Она много дает и мало требует, подвигая даже неповоротливые корпорации на инвестирование перехода к столь совершенному Linux-дистрибутиву. Да, у Ubuntu 6.06 9-месячная поддержка, и примерно такая же у SLED, но RHEL 5 превосходит их как в серверном, так и в настольном вариантах. Чтобы этого достичь, Red Hat пришлось серьезно заняться тем, что прежде недооценивалось: рабочей станцией.

RHEL 5 сметает все сомнения в прочности положения Red Hat на рынке. Помнит кто-нибудь об Oracle Unbreakable Linux? Опасен ли союз Microsoft/Novell? Нет, нет и нет – позиции RHEL незыблемы! LXF



› *Compiz*, на зависть прочим дистрибутивам для предприятий, придает интерфейсу небывалый лоск.

LINUX FORMAT Вердикт

Red Hat Enterprise Linux 5

Разработчик: Red Hat Inc

Сайт: www.redhat.com

Цена: Server от $349, Advanced Platform от $1499, Desktop от $80 на каждую машину.

Функциональность	9/10
Производительность	9/10
Простота использования	9/10
Цена	10/10

» *Большой скачок RHEL, поднимающий планку для SUSE и Oracle.*

Рейтинг 9/10

Thunderbird 2

Сможет ли *Thunderbird* вырвать клиентов электронной почты из лап Gmail и Yahoo? **Энди Ченнел** приковался к компьютеру, чтобы оценить возможности…

Вкратце...

» Почтовое сопровождение *Firefox*, теперь с поддержкой RSS. См. также: *Kontact*, *Evolution* или *Opera*.

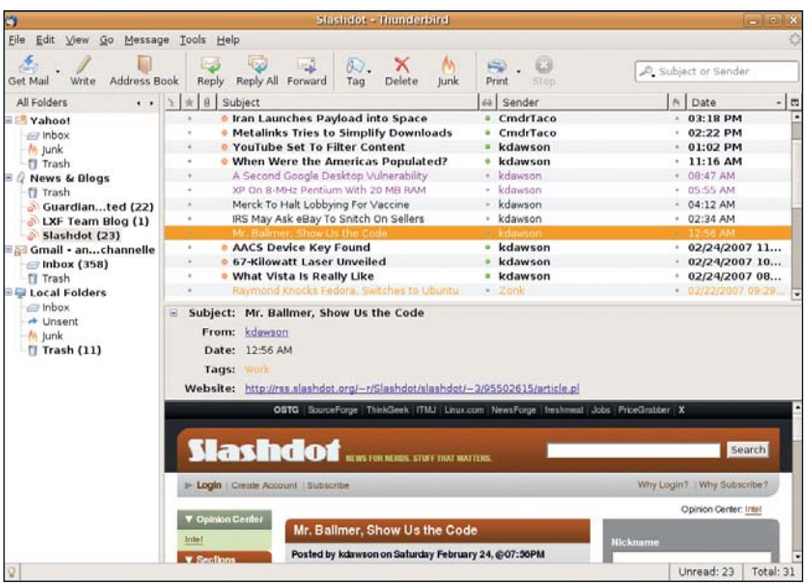
Похоже, весь мир против *Thunderbird*. Компьютеры поставляются с готовыми почтовыми клиентами, и две превосходные программы – *Kontact* и *Evolution* – не дают забыть о себе большинству Linux-пользователей. Больше того, новейшие web-службы упрощают доступ к почте настолько, что она выглядит частью рабочего стола. Есть ли шанс у *Thunderbird*, кросс-платформенного почтового напарника *Firefox*?

Установив вторую версию, мы обнаружили, что возможности *Thunderbird* теперь включают настройку новостных групп и RSS-лент, пометку сообщений, подобную используемой Gmail, приличные функции ведения дискуссий и улучшенные инструменты управления многочисленными учетными записями. Возможности настройки учетных записей расширены настолько, что подключиться теперь можно к POP, IMAP или Gmail, а это дает возможность импортировать новые сообщения от всех провайдеров и почтовых серверов, какие только есть в природе. Не здорово, правда, что первоначальная настройка не позволяет оставлять копии сообщений на сервере. Такая возможность есть, но доступна лишь после входа в готовую учетную запись.

На первый взгляд, кроме дизайна, ничего не изменилось, но серьезные новшества все же есть. Появились приметная кнопка Tag (Пометить) и новый набор кнопок на панели папок, переключающих вид панели в разные состояния. Это полезно, когда панель начинает загромождаться – так происходит у тех, кто часто читает RSS-ленты. Есть и скрытые нововведения, упрощающие работу с большим числом учетных записей: например, установка общей папки «Отправленные» (аналогичной возможности для «Удаленных», впрочем, нет).

> «Приятно видеть поддержку RSS-статей прямо в панели сообщений.»

Новшества показывают, что разработчики *Thunderbird* решили вырваться из болота стандартных почтовых клиентов. Приятно видеть поддержку RSS-лент прямо в панели сообщений (хотя технология RSS, бесспорно, больше подходит для использования в браузере, чем в почтовом клиенте), а возможность добавлять пометки делает просмотр RSS чрезвычайно удобным.



» **Пометки *Thunderbird* помогут, когда сообщения почты и новостей загромоздят окно. Слева – сообщения, импортированные с Yahoo и Gmail.**

Полезные пометки

Пометкам можно задавать цвета, благодаря чему упрощается отметка и поиск важных сообщений. Было бы еще лучше иметь возможность помечать несколько учетных записей и RSS-лент, а затем просматривать отмеченное в одном окне. Можно создать фильтр сообщений, учитывающий такие критерии, как отправитель, ключевые слова или тема, и складывать соответствующие сообщения в отдельную папку, но режим просмотра по пометкам особенно пригодился бы.

Новый способ пометки хорош для почты, но для RSS просто неоценим. Тот, кто просматривает ленты в поисках каких-либо сведений – например, студенты – легко сможет построить библиотеку статей по определенному предмету из любого количества источников, а затем собрать их в одном месте для просмотра.

Прежние органы контроля почты, как всегда, надежны. Это комбинация простого и быстрого ввода пользовательских данных для настройки байесовских фильтров и поддержка *SpamAssassin* или *SpamPal* для закрепления результата. По надежности это близко к возможностям Gmail и значительно превосходит спам-оборону Yahoo.

Единственное, чего не хватает *Thunderbird* – это интегрированного управления временем и информацией (календари, напоминания, заметки и т.п.): все, чем так удобны *Evolution*, *Kontact* и все приличные web-службы. Грядущая добавка *Sunbird* призвана исправить положение, но это будет еще не скоро, а ведь конкуренты не дремлют.

Несмотря на все недостатки, *Thunderbird* нельзя сбрасывать со счетов. Для тех, у кого много почтовых учетных записей, это превосходное решение. Может быть, еще нужнее программа будет корпоративным пользователям, нуждающимся в межплатформенной альтернативе *Outlook Express*, ведь надежный почтовый клиент подобрать не так-то просто. LXF

LINUX FORMAT Вердикт

Thunderbird 2

Разработчик: The Mozilla Foundation
Сайт: www.mozilla.com/thunderbird
Цена: Бесплатно под Mozilla Public License

Функциональность	9/10
Производительность	9/10
Простота использования	9/10
Документация	7/10

» *Фантастический читатель почты и RSS. Заинтересует корпоративных пользователей, но не тех, кто живет в Сети.*

Рейтинг 9/10

Rosegarden 1.5

Система нумерации версий более разумная, кодовая база переработана, но розы-то в этом саду остались? **Грэм Моррисон** принюхивается.

Вкратце...

» MIDI- и аудиосеквенсор с поддержкой программного синтезатора и эффектов реального времени. См. также: *Ardour* и *Muse*.

За два года, прошедшие со времени выхода последней версии *Rosegarden* (почему-то названной версией 4), разработчики хорошо потрудились над добавлением функций и оптимизацией кода. Заодно они создали Fervent Software и выпустили коммерческий Linux-дистрибутив под названием Studio To Go. *Rosegarden* составляет сердцевину Studio To Go, и по функциональности это открытое приложение ближе всего подходит к уровню программ, используемых мировыми лидерами звукозаписи: *Steinberg Cubase*, *Logic Pro* от Apple и *Digital Performer* от MOTU.

Rosegarden использует окно аранжировки: вертикальные дорожки, содержащие аудио- или MIDI-данные, располагаются слева, а сегменты данных каждой дорожки растягиваются вправо. Каждая дорожка обычно – отдельный инструмент или запись, а каждый сегмент – музыкальный фрагмент. Окном аранжировки оно называется потому, что в нем можно свободно перетаскивать блоки влево и вправо по времени, вверх и вниз между инструментами, эффективно меняя «аранжировку» музыкального фрагмента. *Cubase*, *Logic* и *Digital Performer* встраивают в такое окно функции правки, позволяющие редактировать MIDI-ноты и менять громкость фрагментов. *Rosegarden* этого не умеет (пока), но в новой версии переключение между дорожками и управление инструментами значительно упростилось.

Особенно впечатляет программный синтезатор. Мы впервые видим DSSI-инструменты, так хорошо интегрированные в музыкальное приложение для Linux. Избранные инструменты и настройки хранятся вместе с проектом, поэтому обмениваться материалами с другими пользователями теперь можно без долгих переговоров о настройке Jack-соединений. Параметры инструментов устанавливаются с помощью точных органов управления, а звуковые дорожки и треки синтезатора могут использовать LADSPA-эффекты с панели параметров инструментов в реальном времени. Работа со звуком по-прежнему перекладывается на какой-либо внешний редактор, например *Audacity*, но матричное отображение для редактирования MIDI-нот заметно улучшилось. Теперь работать с нотами в *Rosegarden* – одно удовольствие.

Нету розы без шипов

Главных дополнений два. Первое – автоматическое преобразование аудио-фрагментов. Идея такая: если есть фрагмент длиной в один такт, а его нужно растянуть на два, нажатие Ctrl с перетаскиванием границы фрагмента автоматически подгонит звук внутри фрагмента для соответствия новой продолжительности. Масштабируя фрагмент, *Rosegarden* перерабатывает внешним редактором (*sox* или *sndfile-resample*) весь аудиофайл. В зависимости от размеров клипа, это может занять от нескольких секунд до убийственных для производительности нескольких минут. Лучше было бы масштабировать звук только в пределах фрагмента, именно так поступают коммерческие соперники *Rosegarden*.

Мутная аудиоконверсия

Второе заметное дополнение – прозрачная аудиоконверсия. К несчастью, импорт звука неинтуитивен, для него нет пункта меню. Необходимо перетащить аудиофайл в окно аранжировки, и *Rosegarden* запустит фоновый процесс для его конвертации с помощью внешнего скрипта. Если у вас нет подходящего инструмента – скажем, *oggdec* – *Rosegarden* безмолвно провалит импорт.

Рассматривая *Rosegarden* в LXF65, мы заключили, что, при всех недостатках, это лучшая аудио/секвенсорная прогамма на Linux. Теперь мы вдвойне в этом уверены, ведь некоторые из недостатков уже исправлены. Внешние звуковые редакторы и программные синтезаторы отныне выглядят «полноправными гражданами», а не примитивными дополнениями к MIDI-секвенсору. С учетом нововведений 1.5, и если разработчики сохранят прицел на интуитивность – мы ждем не дождемся версии 2.0! LXF

Исследуем интерфейс Rosegarden

Микшер
Аудио- и MIDI-микшер собирают важнейшие консольные органы управления в единственном окне.

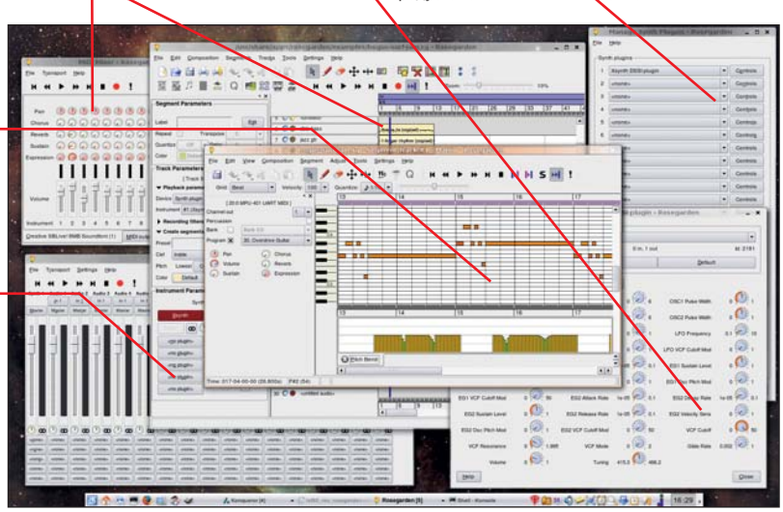
Нотная матрица
Отображает продолжительность нот и упрощает их добавление.

Информация о дорожке
Пользуйтесь этой панелью для добавления эффектов реального времени и управления MIDI.

Окно аранжировки
Позволяет свободно перемещать фрагменты аудио и MIDI.

Редактор синтезатора
Каждый инструмент имеет собственный редактор и менеджер предустановок.

Панель синтезатора
Программные синтезаторы загружаются и управляются в своем окне.



LINUX FORMAT Вердикт

Rosegarden 1.5

Разработчик: Chris Cannam, Richard
Сайт: www.rosegardenmusic.com
Цена: Бесплатно под GPL

Функциональность	9/10
Производительность	6/10
Простота использования	7/10
Документация	7/10

» *Интуитивно и достаточно гибко для мира современной музыки.*

Рейтинг 8/10

Gnome 2.18

KDE 4 на горизонте не видно, других соперников нет, поэтому Gnome переключился на круиз-контроль, полагает **Пол Хадсон**.

Вкратце...

» Популярная, доступная в освоении рабочая среда, антитеза KDE. См. также: KDE, *Xfce*.

Грэм считает...

«Разработчики Gnome прочно стоят на ногах. Твердая уверенная поступь, регулярный цикл обновления. Но новаторства слишком мало, и итог не вдохновляет».

Это не революционная версия Gnome (а много ли таких было?). В черновике этой статьи мы не побрезговали даже упоминанием о том, что *Gnibbles* – Gnome'овский клон знаменитой «змейки» – теперь поддерживает многопользовательскую игру через интернет благодаря Gnome Gaming Zone. Не будучи крупным релизом, версия все же не лишена новизны: обновлен дизайн Центра управления; *Glade 3* и *NetworkManager*, наконец, попали в состав базового дистрибутива, да и *Tomboy* получил несколько ценных дополнений.

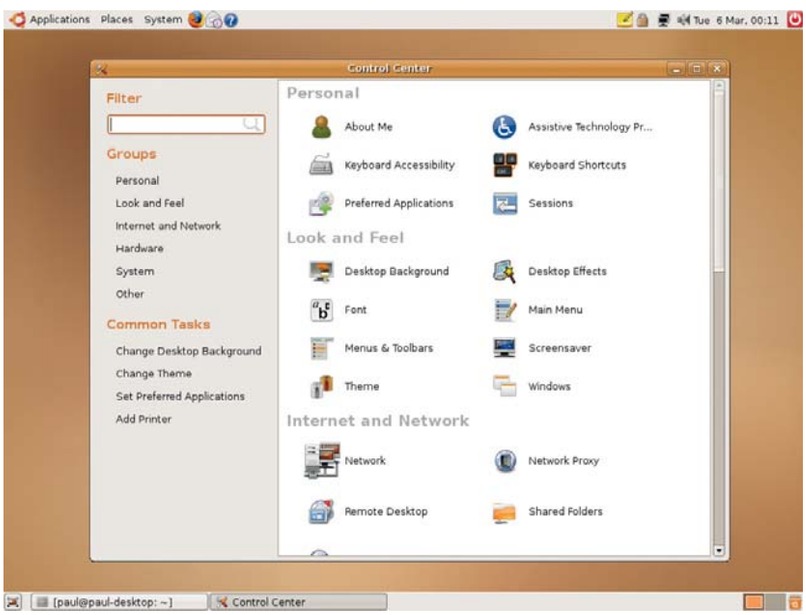
Больше всего нам понравился (хотя это просто милая мелочь) новый интерфейс *Gnome Screenshot*. Нажатие Print Screen вызывает привычное диалоговое окно, но если запустить *Gnome Screenshot* из меню Applications (Программы), то появится новый интерфейс, где можно выбирать между захватом всего экрана или только активного окна, а также устанавливать время, через которое нужно сделать снимок. Конечно, *KSnapshot* научился этому уже давно, но Gnome такой функции остро недоставало. Особенно радует тот факт, что окно снимается автоматически, а заголовок используется в качестве имени файла. Пока это доступно только Gnome.

Все вместе, не спеша

Прежний Центр управления Gnome фактически являлся меню Preferences (Параметры), переведенным на иконки, но Gnome 2.18 представляет новый Центр, разделенный по категориям – он впервые увидел свет в SUSE 10.2. Он содержит и значки параметров, и значки администрирования, и большинство пользователей, скорее всего, заблудится среди 40 иконок окна Центра управления. Слева есть поле поиска по названию, а также список задач, например, добавление принтера – это немного снижает сложность, но над упрощением еще предстоит поработать.

> «Seahorse, встроенная в Nautilus, шифрует и дешифрует файлы».

Мы упомянули об обновлении *Tomboy*; особенно нам нравится новая функция прикрепления особо важных заметок к выпадающему меню, так что они не теряются среди множества других. Теперь можно создавать маркированные списки, начав новую строку со звездочки или дефиса, что экономит несколько мышиных щелчков.



› Новый *Центр управления* Gnome распределяет параметры по категориям – выглядит мило, но без фильтра ничего не найдешь.

Vino – VNC-система удаленного рабочего стола Gnome – теперь сообщает о подключении к вашей машине другого пользователя: это еще одна функция, которой нам давно не хватало. Когда кто-нибудь подключается, появляется маленькое всплывающее окно с указанием IP-адреса, и чтобы отказать в подключении, достаточно щелкнуть правой кнопкой мыши на значке *Vino*.

Старый пес, старые штуки

Если вы ищете в Gnome сногсшибательных новшеств, то вы ищете не там. Фактически, даже два самых заметных дополнения – *Glade 3* и *NetworkManager* – уже давно засветились в Gnome-дистрибутивах. Единственная подлинно новая программа – это *Seahorse*, графический интерфейс к *GPG*, ловко интегрированный в *Gedit* как расширение. Программа позволяет создавать *GPG*-подписанные сообщения прямо в *Gedit*. *Seahorse* встроена и в *Nautilus*, а это значит, что можно шифровать, дешифровать и подписывать файлы правым щелчком мыши и вводом пароля.

Как вы уже поняли, эта версия не очень-то богата функционально. Основной объект нашей критики – полное отсутствие ярких новинок. Если вас волнует не только новый номер, усовершенствования в Gnome 2.18 вы разыщете не сразу. Когда *Sound Juicer* сообщит вам о том, что установлен новый модуль *GStreamer*, и теперь вы можете кодировать дорожки прямо в MP3; или когда вы впервые заметите, как *Eye of Gnome* автоматически поворачивает фото (поняв по тэгу EXIF, что оно снято в портретном, а не в пейзажном стиле), обновление вас порадует. Но, если команда Gnome хочет достичь чего-то большего, ей следует задуматься о более крупных переменах. LXF

LINUX FORMAT Вердикт

Gnome 2.18

Разработчик: The Gnome Project
Сайт: www.gnome.org
Цена: Бесплатно под GPL

Функциональность	8/10
Производительность	8/10
Простота использования	9/10
Документация	9/10

» *Gnome 2.16, но несколько освеженный. По крайней мере, безвреден.*

Рейтинг 8/10

Genius Look 316

Web-камера с готовой поддержкой Linux?
Ник Вейч еще поглядит…

Вкратце...

» Web-камера – минное поле для обеспечивающего ее поддержку в Linux, ведь изготовители меняют чипсеты постоянно.

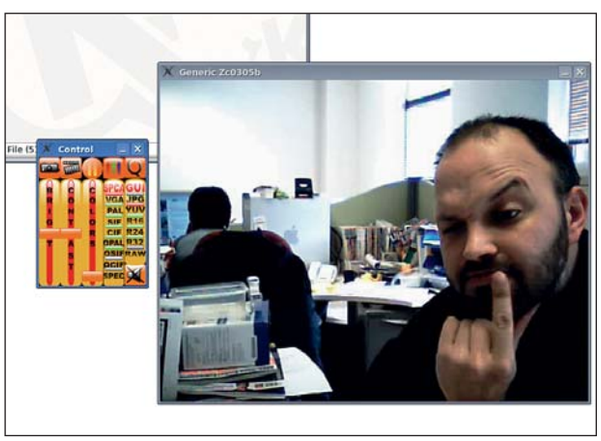
Наш обычный подход к обзорам оборудования таков: если нет Linux-драйверов от изготовителя – то и Бог с ней, разве что устройство уж очень классное. В самом деле, с какой стати нам поддерживать изготовителя, который не собирается поддерживать нас? Но коль скоро объявлено, что web-камера Genius Look 316 поставляется с Linux-драйверами, мы ее не упустим.

У Look 316 стандартные размеры и форма (мячик для гольфа), ставшие уже привычными для всего мира. Оснащение такое: кнопка «активации», обычно используемая Windows-приложениями для запуска программ захвата видео; вручную фокусируемая стеклянная линза, которую можно перемещать вверх-вниз; удобный скользящий кронштейн для крепления камеры на верху монитора или ноутбука. Вот и первая неприятность: клипса кронштейна максимально раздвигается всего на 2 см, и его не прицепишь даже на Майка, не говоря уже о тех местах, где камера действительно нужна.

Глянцевая упаковка обещает 1.3-мегапикселя на статическое изображение, но само устройство ограничено разрешением VGA – «высокое разрешение» достигается лишь шаманством DirectX под Windows (или обработкой картинки в *Gimp*). На самом деле вы получаете возможность передавать VGA-картинку (по 15 кадров в секунду) через стандартный интерфейс USB 1.1, то есть ничего впечатляющего, хотя использование стеклянной линзы взамен пластика облагородило изображение. Кстати о звуке: его нет.

Если вы все же вынуждены использовать Look 316 как web-камеру, то понадобится *Ekiga* (прежде известная как *GnomeMeeting*) или пакет SPCA, имеющийся на том же сайте, откуда мы добыли рабочие драйверы (см. *«Сказка о двух драйверах»*, ниже). SPCA делает экранные снимки, служит сервером изображений (хотя *CamE* тоже годится) и предоставляет (капризный) GUI для управления камерой. Для устройства заявлена частота 30 кадров в секунду, но – то ли из-за нашего ПО, то ли из-за драйверов – достичь ее не удалось.

Камера недорогая, да и качество неплохое. Но заявление на упаковке о поддержке Linux – дезориентация, если не сказать больше. Поддерживается только некий экзотический вариант ядра трехлетней давности. Мы с трудом вспомнили, когда выходил последний дистрибутив на ядре 2.6.5 по умолчанию – а ведь в Genius не решились бы заявить о поддержке Windows, если бы камера работала лишь на Windows ME. LXF



› Ради Genius-устройства редактор прикинулся гением.

› Мы включили свой Genius для исследования Временной Относительности в Экранном Пространстве.

Сказка о двух драйверах

Для устройства заявлена поддержка в Linux. Есть даже маленький логотип на упаковке, подтверждающий это. Однако, обозрев инсталляционный диск, мы обнаружили какой-то скрипт и единственный прекомпилированный бинарный драйвер для… ядра 2.6.5. Ладно, и на том спасибо – вернемся назад, в 2004, и попробуем, трудно, что ли? Ох, трудно: ни исходника, ни намека, как поступать тем, кто сдуру обновил ядро за последние три года. Посещение www.geniusnet.com успеха не имело. В спецификациях особо оговаривается ядро 2.6.5, что опрометчиво можно принять за «2.6.5 и выше».

Устройство не удалось заставить работать и с открытыми драйверами, которые подвесили несколько Linux-машин с Ubuntu! Печальная история повторялась до тех пор, пока мы не догадались взять Fedora Core 6 и загрузить пакеты с http://mxhaard.free.fr для драйвера spca5x к новым ядрам, который действительно заработал. Вот и все. Хотя ничего больше не ломалось, большинство сопутствующих программ (*Xawtv*, *MPlayer* и т.п.) не смогло работать с устройством напрямую, справились с этим лишь немногие (например, *Ekiga* и *CamE*).

LINUX FORMAT Вердикт

Genius Look 316

Разработчик: Genius
Сайт: www.geniusnet.com
Цена: около $30

Функциональность	5/10
Производительность	8/10
Простота использования	2/10
Цена	5/10

» *Исполнение неплохое, но с Linux-поддержкой неважно.*

Рейтинг 3/10

Komodo IDE 4.0.2

За свои почти $300 этот пакет должен быть уж очень хорош, чтобы **Майк Сондерс** оставил испытанный *Nano*…

Вкратце...

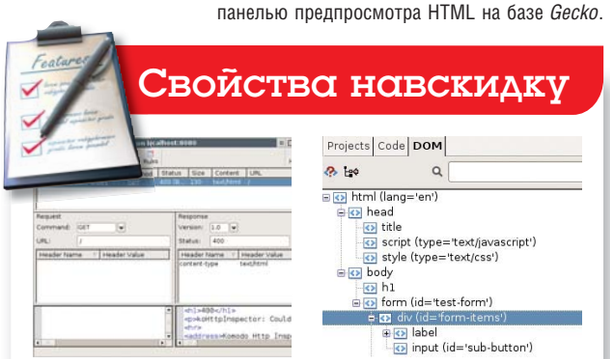
» Программа для разработки web-сайтов с поддержкой нескольких языков. См. также: *Quanta* или *Bluefish*.

Помните Mozilla Application Framework? Когда в 1998 был открыт исходный код *Netscape*, программисты надеялись превратить браузер в платформу для написания программ, распространяемых через Интернет. Эти надежды не сбылись: кроме немногочисленных демо-версий (вроде *Mozilla Amazon Browser*, http://faser.net/mab) из проекта ничего не вышло. То есть, пока не выходило – а теперь встречайте *Komodo* от ActiveState, среду разработки, построенную на кодовой базе *Mozilla* [это гротеск: проект *Komodo* существует около пяти лет, – прим.ред.]!

Всем известно, что *Mozilla* – даже в своих *Firefox*- и *Thunderbird*-ипостасях – не особо шустрая программа: это отразилось и на *Komodo*. На нашей тестовой машине с 1,7 ГГц среда загружалась восемь секунд из холодного состояния, а во время работы безбожно тормозила. Поэтому минимальные системные требования (233 ГГц) выглядят безудержно оптимистичными: 1.5 ГГц, 256 МБ ОЗУ и 230 МБ свободного места – так было бы вернее.

Конек *Komodo* – поддержка множества языков программирования: инструментов кодирования под Linux не счесть, но лишь *Komodo* может похвастаться собранием CSS, HTML, JavaScript, Perl, PHP, Python, Ruby, Tcl, XML и XSLT под одной крышей. Это покрывает основные нужды web-разработки, а значит, *Komodo* можно использовать для написания Ajax-приложений – очко в его пользу.

Основной компонент *Komodo* – это редактор (первоклассный). Он включает обязательное сворачивание кода, подсветку синтаксиса и вкладочное редактирование, наряду с эмуляцией клавиатурных комбинаций *Emacs* и *Vi* и панелью предпросмотра HTML на базе *Gecko*.



› **При 1024x768 главное окно захламлено, но можно изменять размеры панелей и удалять их.**

Но самая полезная функция – проверка синтаксиса на лету: во время программирования *Komodo* проверяет код и подчеркивает ошибки красной волнистой линией.

Широта стиля

Кроме того, *Komodo* не забывает о стилистической стороне кода – например, рекомендует атрибуты к HTML-тэгам во время набора. Эта функция работает превосходно и сберегает кучу времени на чтении сообщений об ошибках и поисках неверных строк. Редактор умеет завершать код и показывает всплывающие окошки с краткими описаниями функций, методов и их параметров, а во время отладки можно устанавливать точки останова и проверять код построчно.

Интеграция с системами контроля версий очень удобна для коллективной работы и охватывает *CVS*, *SVN* и *Perforce*. Среди приятных особенностей – *RX Toolkit*, облегчающий создание регулярных выражений, и Toolbox, с которым можно подключить инструменты командной строки (например, подсчет слов *wc*) к рабочей среде редактора.

Особой похвалы заслуживает документация *Komodo*, в которой для ясности широко используются снимки экрана и даже есть мини-учебники по PHP, Perl, Python и Ruby. Стыдно, однако, что возможности поиска в документации ограничиваются заголовками тем, полнотекстового поиска нет.

Естественно, цена *Komodo* не позволяет ей стать основным инструментом web-разработчика. Того, кто программирует на трех-четырех языках, интеграция, возможно, обрадует, а для Perl-хакера, например, комплект избыточен и слишком дорог. *Komodo* очень богат функционально, и на нашем компьютере вел себя стабильно – хотя интерфейс чересчур громоздкий, даже для машин среднего класса. С другой стороны, это прекрасный пример надежной, функционально богатой среды web-разработки, и вы можете взять 21-дневную ознакомительную версию на сайте ActiveState. LXF

Свойства навскидку

HTTP-инспектор
Встроенный прокси-сервер, работающий через порт 8080, упрощает анализ состава HTTP-запросов.

DOM-браузер
Панель отображает XML- или HTML-документ в виде дерева для навигации между элементами.

LINUX FORMAT Вердикт

Komodo IDE 4.0.2

Разработчик: ActiveState
Сайт: www.activestate.com
Цена: $295

Функциональность	9/10
Производительность	5/10
Простота использования	8/10
Цена	6/10

» *Дорогой и медлительный, но упакован под завязку.*

Рейтинг 7/10

Технология счастья

SUNRADIO.RU

сетевое радио под ключ на базе Linux

новое будущее вашей компании

pr@sunradio.ru +7 812 955 76 70 www.sunradio.ru

КОРПОРАТИВНЫЕ СЕРВЕРЫ
И СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ

(812) 327-5960
(495) 232-9230
info@trinitygroup.ru

TRINITY
CORPORATE IT PROJECTS

- Анализ существующей IT инфраструктуры
- Разработка технического задания
- Проектирование, монтаж, внедрение
- Комплексное управление IT инфраструктурой
- Катастрофоустойчивые решения

Серверы

под Linux
FreeBSD
Solaris x86

для баз данных, интернет шлюзов, WEB-приложений, кластеры для научных расчетов

Мы делаем бизнес успешным

www.trinitygroup.ru

Информационные технологии от экспертов

Blender 2.43

Разделавшись с учебником «Blender для начинающих», **Андрей Прахов** сумел, наконец, выкроить минутку, чтобы взглянуть на новую версию.

Вкратце...

» Кроссплатформенный открытый 3D-редактор, отличающийся широкими возможностями в моделировании и анимации, сравним по мощности с 3D MAX. См. также: *Ayam*, *Moonlight 3D*.

Февраль текущего года ознаменовался примечательным событием в мире Open Source, которое откликнулось в душах истинных художниках – вышла очередная версия мощного, бесплатного 3D-редактора *Blender*. Пусть со стороны эта фраза кажется излишне напыщенной, но она того стоит! Последняя версия программы имела номер 2.42a. Вроде бы совсем мизерное увеличение, но разработчики *Blender*, как известно, отличаются стойким пренебрежением к резким революционным скачкам, предпочитая сочетать плавные изменения в нумерации релизов со значительными изменениями в программе. Вот и сейчас за этой цифрой скрывается немало новых возможностей и доработок.

Новинки из области редактирования

Как известно, процесс создания модели заключается в скрупулезном, тонком редактировании вершин, граней объекта. Хотелось бы что-нибудь этакого воздушного, как взмах кисти художника. И это стало возможно в *Blender*! Рассматриваемая версия представляет вашему вниманию новый режим редактирования – Sculpt Mode.

Чтобы стало понятно, что это такое, вообразите себя на мгновение скульптором, ваяющим очередной шедевр. Из безликой массы глины под точными движениями зарождается нечто, воплощающее идеи мастера. Где-то что-то вдавливается, округляется или наслаивается, и вот момент – срывается покрывало, и то, что было просто глиной, обретает черты, живет. Но все это лирика. В действительности, программа предоставляет достаточно небольшой набор типов кистей с определенными функциями. Кажется, не Бог весть что, но теперь можно сглаживать и выдавливать движениями мыши, а если вы обладатель графического планшета, то работа упрощается в разы.

Ранее для сглаживания объекта можно было использовать модификатор Subsurf. В целом, со своей задачей он справляется неплохо. Но как быть, если при редактировании необходимо не столь сглаживание, сколь детализация? Конечно, использовать функцию Subdivide! А если потребуется откатить на более ранний уровень детализации… В общем, мрак.

Новый инструмент Multires предоставляет возможность свободного выбора нужной детализации объекта с сохранением редактирования на всех уровнях. То есть, если мы что-то отредактировали на начальном уровне – оно автоматически изменится и на более высоком. Причем вы вполне можете работать с низкой детализацией, а обрабатывать по максимуму. Это, как оказалось, очень и очень удобно.

› **Режим скульптора: так действовал Роден, создавая своего Мыслителя.**

Изменения коснулись и самих модификаторов. Во-первых, разработчики переписали почти весь код с нуля и добились лучшей интеграции с остальными частями программы. Теперь их расположение в стеке можно менять произвольно, не опасаясь за сохранность проекта. Кроме того, добавлены три новых модификатора: Displace, EdgeSplit, UVProject.

Displace дает возможность дизайнеру создавать реалистичные движения таких объектов, как облака, водная зыбь, т.е. всего, где нужен эффект колебания. Технически это создается перемещением вершин в соответствии с RGB-составляющей текстуры. Формула «RGB -> XYZ», естественно, действительна для каждой вершины.

Еще один очень интересный и удобный модификатор – UVProject. Пожалуй, с появлением этого инструмента, проблемы позиционирования текстуры на объекте будут решены. Если вы когда-нибудь работали с *3D Max* и его редактором *UVWs*, то можете представить сложность и громоздкость этой процедуры. С появлением модификатора UVProject в *Blender* задача значительно упрощается. Теперь за правильное наложение текстур отвечают специально настроенные камеры, поворотом или движением которых можно двигать и текстуры. Причем количество камер произвольно, и каждую из них можно нацелить на нужную часть объекта.

Больше движения – больше естественности

Примечательной чертой *Blender* является хорошая имитация некоторых физических явлений. В новой версии был доработан fluid simulator, часть программы ответственная за взаимодействие объектов с жидкой средой. Чтобы лучше понять, о чем идет речь, представьте каплю дождя, упавшую в воду, или корабль, рассекающий волны. Для естественности происходящего необходимы весьма длительные и сложные расчеты. В прежней версии *Blender* их результат выглядел не слишком удачно. Теперь fluid simulator обзавелся возможностью улучшить качество картинки за счет нового алгоритма обработки mesh-объекта жидкости. По сути дела, применяется модификатор subsurf, учитывающий особенности жидких частиц. А если прибавить появившуюся технологию внедрения анимации объектов, таких, как арматурные, то мы

получаем эффективный механизм демонстрации необычных, но естественно выглядящих жидких эффектов. Так, к примеру, можно создать человека, созданного из воды.

Серьезные изменения коснулись и работы со скелетной анимацией. Не секрет, что наиболее сложное в этой области – добиться естественности при имитации движения. Особенно если есть необходимость смоделировать походку «живого» объекта по сложной пересеченной местности. Так, идущий наверх человек нагибает корпус немного вперед и как бы подтягивает ноги за собой. А если спускается, то «ножки первыми бегут по дорожке». Еще необходимо и следить за тем, чтобы объект не проваливался или повисал в воздухе. Вот уж истинные мучения аниматора! Поэтому, думаю, пользователи *Blender* с истинным удовлетворением встретят новые возможности программы по автоматизации привязки к местности. Проще говоря, ничто теперь не мешает вам использовать обычный контур Curve Path для выстраивания маршрута движения персонажа, а уж программа сама подстроит движения костей оптимальным образом.

И, наконец-то, интересные новшества ждут нас во встроенном нелинейном редакторе. Во-первых, появилась возможность трансформировать объект непосредственно на монтажном столе. Теперь вы сможете, выбрав пункт меню Effect -> Transform, воплотить любые задумки движения, ротации и масштабирования, воспользовавшись специальным окном. Необходимо отметить начало и конец эффекта, настроить пространственные изменения, и вуаля – вы получаете готовую картинку. Эта идея вплотную приближает *Blender* к нелинейным видеоредакторам.

Из приятных мелочей, можно отметить появление генератора цвета. Раньше приходилось загружать графический файл с подходящей окраской. Теперь можно выбрать нужный цвет из палитры – что может быть проще?

Но самое интересное из нововведений – это возможность изменения скорости проигрывания. Воспользовавшись Ipo Curve Editor можно играть с замедлением или ускорением в любых масштабах и вариациях. Это значительно упрощает работу над конечным результатом и дает немалую фору творческому мышлению.

Конечно, все нововведения описать в одной статье невозможно, да и не нужно. Важно то, что наш любимый редактор не стоит на месте, а развивается, обогащаясь интересными и полезными функциями. За дополнительной информацией обращайтесь на официальный сайт www.blender.org. LXF

› Встроенный видеоредактор тоже похорошел.

LINUX FORMAT Вердикт

Blender 2.43

Разработчик: Blender Foundation
Сайт: http://www.blender.org/
Цена: Бесплатно по лицензии GPL

Функциональность	10/10
Производительность	9/10
Простота использования	6/10
Документация	7/10

» Единственный профессиональный выбор для 3D моделлеров и аниматоров, работающих в Linux.

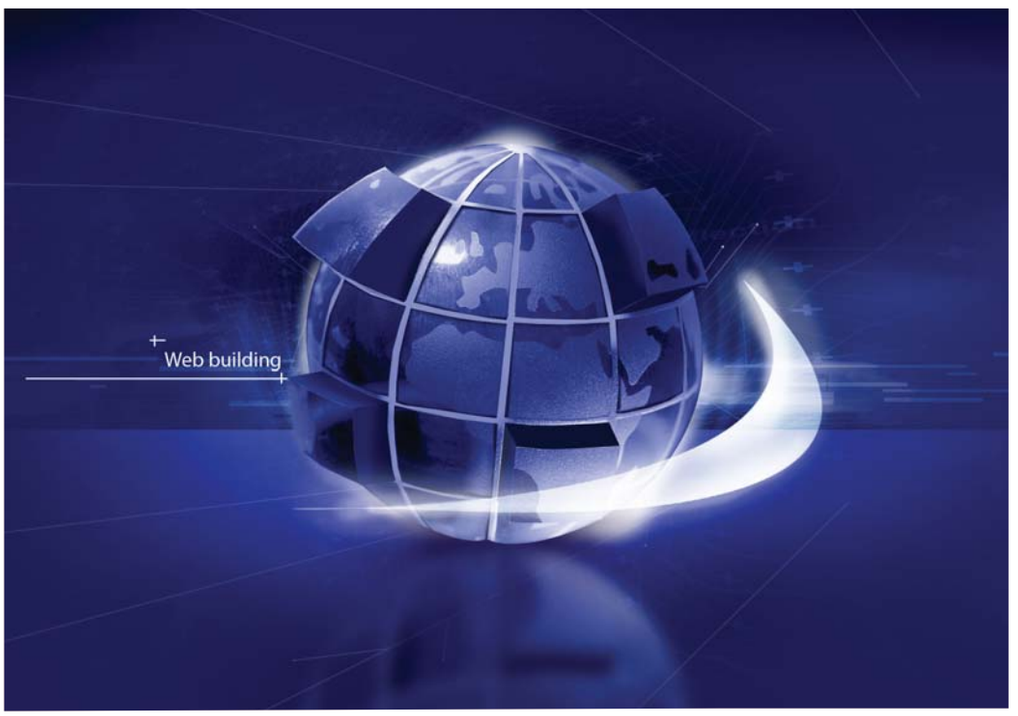
Рейтинг 9/10

Сравнение

Каждый месяц мы сравниваем тонны ПО — а вы можете отдыхать!

Web-редакторы

В шоке от перспективы создавать web-сайт? **Дэвид Каулсон** знает HTML-редакторы, делающие эту работу пустячной.



Paul Blachford

Про наш тест

Мы опробовали каждый редактор на некотором реальном web-контенте, затем воспользовались им для создания нескольких страниц с нуля. Вот на что мы обращали внимание:

Простота использования. Мы пользуемся редакторами, потому что *Vi* просто слишком сложен для большинства людей. Мы оценивали интерфейс пользователя, а также возможности для опытных пользователей.

Поддержка языков программирования. Создание статической страницы с анимированными тостерами – это 1995-й год. Мы проверяли, могут ли редакторы работать с CSS, PHP и Perl.

Генерируемый код. Каждый знаком с... гм... качеством HTML, которое производит *Microsoft Word*. Мы сравнивали код, созданный этими редакторами, с тем, что действительно требуется для отображения нужной страницы.

Наш выбор

OpenOffice.org Web с. 24
Nvu с. 22
Quanta Plus с. 23
Quanta Gold с. 23
Bluefish с. 21
SeaMonkey Composer с. 21
Screem с. 24

В наши дни чуть ли не каждый имеет свое «представительство» в Интернете (по крайней мере, временами так кажется), будь то созданный с нуля сайт или шаблонное CMS-приложение. И хотя появление Web 2.0 принесло с собой сложнейшие сайты, использующие Flash и JavaScript, большинство web-приложений все еще сильно зависит от традиционного HTML и кода, выполняющегося на стороне сервера.

Те из нас, кому случалось создавать сайты и писать приложения, часто сталкивались с необходимостью скрестить функциональный код с оформительской частью всего за полдня. Немало опытных web-дизайнеров, добиваясь идеала, вручную создают и подчищают HTML и CSS, но большинство из нас предпочли бы щелкать мышью да перетаскивать фрагменты страниц. *Emacs* и *Vi*, предлагающие подсветку синтаксиса для кода на PHP, Perl и HTML, безусловно, главные претенденты на основную массу приложений. Продуктивно? Без сомнения. Удобно для пользователя? Не особо.

Здесь-то и выручают web-редакторы. Подобно настольным издательским системам, они разработаны, чтобы помочь создавать web-страницы, интегрируя все части проекта в одно приложение. Нужно создать слой, изменить код или даже вручную что-то подредактировать? Вот инструменты для вас. Со времен выхода *Netscape Composer* рынок web-редакторов разросся и сейчас предлагает множество приложений для удовлетворения нужд всех пользователей.

Наиболее существенная функция, связанная с удобством web-редакторов – это наличие режима WYSIWYG (What You See Is What You Get – что видите, то и получаете). Этот интерфейс позволяет вносить изменения в режиме реального времени, с просмотром того, как страница будет выглядеть в браузере, а не в текстовом окне или в плохо отрисованной версии сайта. В связке с продвинутыми функциями работы со встраиваемыми изображениями или таблицами можно создавать сложные конструкции без необходимости постоянно сохранять и тестировать страницу. Это значительно сокращает время, необходимое для внесения небольших изменений, плюс позволяет легко подстраивать код, чтобы убедиться, что все выглядит «как в лучших домах Лондона и Парижа».

Bluefish

Функциональный, быстрый и популярный редактор.

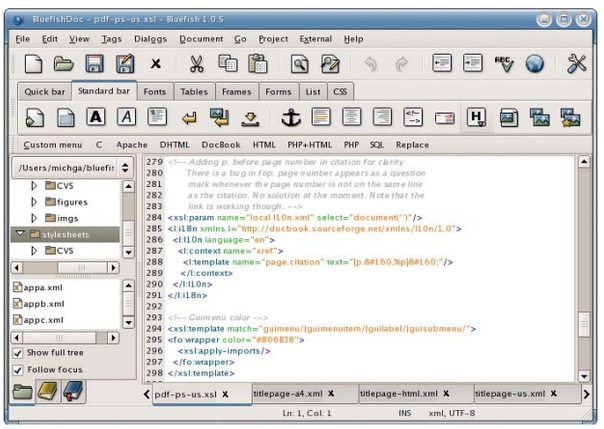
Bluefish – удобный текстовый HTML-редактор для каждого, кому нужно выдать на-гора кучу web-контента без лишней мороки. Не смущайтесь низким номером релиза 1.0.7: *Bluefish* – зрелое приложение, делающее свое дело быстро и эффективно. Версия 1.0.7 содержит множество функций, способных превратить создание очередной web-статьи в удовольствие. Для начала, есть превосходная подсветка синтаксиса для кода на HTML, Perl, PHP и C, упрощающая жизнь, когда приходится вносить изменения непосредственно в страницы.

Другая опция, которую *Bluefish* предлагает по умолчанию – интеграция с браузерами. Для web-разработчика это очень важно: что за радость проектировать страницу, оптимизированную для одного браузера, лишь для того, чтобы увидеть ее изгаженной в другом? Единственный браузер, отсутствие опции для которого мы заметили, был *Internet Explorer*. Пусть вы не любите *IE*, но при проектировании web-сайта важно убедиться, что он выглядит одинаково во всех браузерах, которые могут использовать посетители – по правде говоря, довольно непрофессионально со стороны команды разработчиков *Bluefish* исключать *IE* из-за личного предубеждения, а не стараться удовлетворить потребности своих пользователей.

Bluefish – действительно очень умелый редактор кода, не говоря уже о том, что это подлинно специализированный инструмент web-разработки. Более того, разработчики *Bluefish* вызывают огонь на себя и продолжают разработку *Bluefish* в самом *Bluefish*. Для большинства языков имеется встроенная справка, позволяющая бегло окинуть взглядом список передаваемых функции аргументов, на случай его изменения (такое нередко происходит в PHP).

К сожалению, *Bluefish* не хватает WYSIWYG-редактора, что делает его менее желанным для неопытных пользователей: это приложение спроектировано заядлыми кодерами для людей с таким же складом ума. Будем надеяться, что в следующем релизе они возьмутся

> «Есть превосходная подсветка синтаксиса для HTML, Perl, PHP и C.»



Код на различных языках в этом *GTK*-редакторе.

Вердикт

RCS

Версия: 1.0.7
Сайт: http://bluefish.openoffice.nl
Цена: бесплатно под GPL

Отсутствует нужная многим web-разработчикам интеграция с браузерами, но это отличное решение для пользователей Perl или PHP.

Рейтинг 6/10

SeaMonkey Composer

Открытый отпрыск редактора Netscape.

Запустив *SeaMonkey Composer*, вы сможете работать в режиме WYSIWYG, позволяющем создавать и редактировать свою web-страницу и просматривать изменения большей частью в визуальном интерфейсе. Мы сказали «большей частью», потому что этот редактор также предоставляет определенный тип визуальной индикации элементов вроде таблиц без границ, давая вам знать, где расположены ячейки таблицы. Режим предпросмотра покажет вашу страницу в том виде, в каком она появится в окне Gecko-браузера. В этом окне также можно выполнять редактирование.

А можно работать и в HTML-редакторе. Он снабжен режимом HTML Source, где вы просматриваете и редактируете код, сгенерированный программой. Еще есть режим HTML Tags, в котором ваша страница отображается, в основном, в формате WYSIWYG, но с подсветкой на экране основных HTML-тегов. Нам это пригодилось при поиске ненужных HTML-тегов, созданных редактором. Есть подсветка синтаксиса, но без поддержки для других языков программирования.

Пользование определенными коммерческими WYSIWYG-редакторами могло создать у вас привычку ожидать множества ошибок при любой проверке вашей страницы на соответствие стандартам W3C. Вы будете приятно удивлены, обнаружив, что код, генерируемый *SeaMonkey*, проходит проверку без ошибок. Или, как минимум, это справедливо для нашей тестовой страницы. Поскольку это была самая настоящая web-страница, отнюдь не старомодная, рискнем утверждать, что *SeaMonkey* в целом генерирует правильный HTML-код.

Будучи WYSIWYG-редактором, *SeaMonkey Composer* позволяет новичкам быстро приступить к работе над своей web-страничкой. Хотя ему недостает форм и абсолютного позиционирования CSS, он способен обходиться без этого, благодаря наличию встроенного режима редактирования HTML-кода.

> «Вы будете приятно удивлены, обнаружив, что код лишен ошибок.»

Ответвлению *Netscape Composer*, новой версии от Mozilla, недостает функций, достойных XXI века.

Вердикт

SeaMonkey Composer

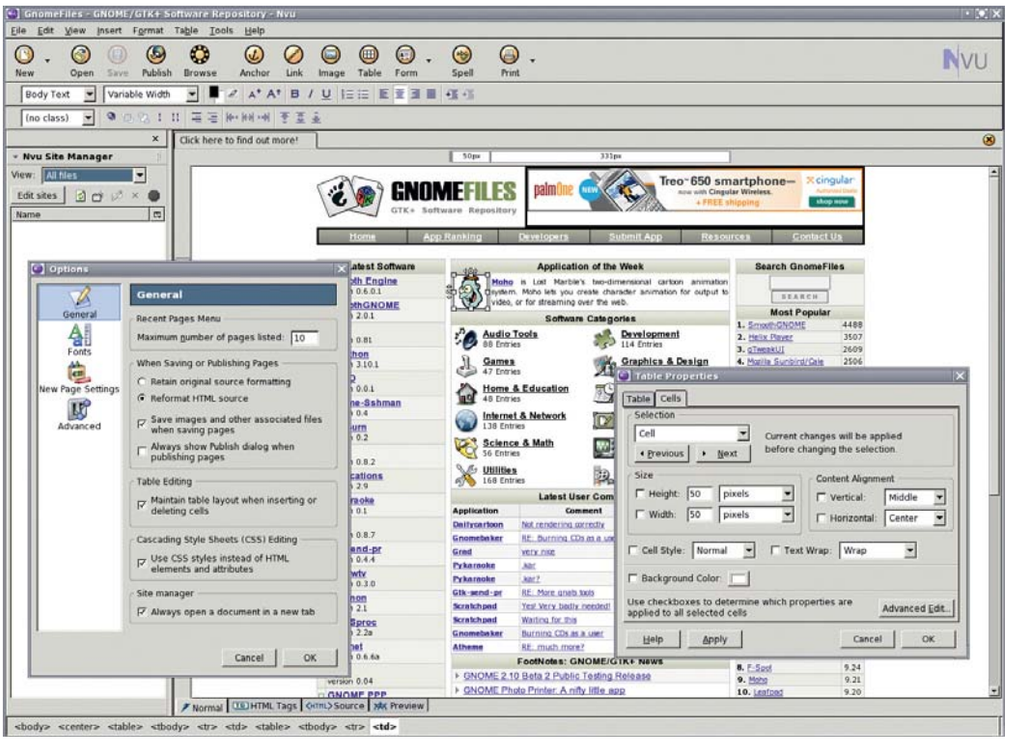
Версия: 1.1.1
Сайт: www.mozilla.org
Цена: бесплатно под GPL

Простой в использовании WYSIWYG-редактор, но без многих продвинутых функций, нужных разработчикам сегодня.

Рейтинг 5/10

Nvu

На базе *Composer*, со спонсором Linspire – но что в нем хорошего?



Nvu построен на кодовой базе *Mozilla Composer*, и если вы дружили с этим древним редактором, навигация в пользовательском интерфейсе *Nvu* будет для вас простой. Вас встретит чистый экран, где можно приступить к созданию своей web-страницы и вставлять графику. Средства, обычно требуемые для создания страницы, вроде вставки ссылок, изображений, таблиц и форм, доступны на панели инструментов, в меню или по «горячим» клавишам; это превращает редактор в гибкий инструмент. Например, новички, которые любят большие, сами за себя говорящие кнопки, найдут полезной панель инструментов, а опытные пользователи, предпочитающие не отрывать пальцы от клавиатуры, смогут работать, используя почти исключительно «горячие» клавиши.

В отличие от *Mozilla Composer*, *Nvu* позволяет создавать весь сайт в WYSIWYG-редакторе, это упрощает выполнение базовых задач, типа вставки изображений и ссылок. Но вы легко можете переключиться из режима визуального редактирования в режим исходного кода (HTML Source), режим предпросмотра (позволяющий увидеть страницу так, как ее отображает движок *Gecko*) или режим *Tags*, показывающий вашу страницу с подсвеченными HTML-тегами. Встроенная поддержка FTP позволит вам загрузить свои страницы на web-хост. Ее полезность ограничивается встроенным FTP-клиентом, которого, впрочем, вполне достаточно для основных задач загрузки страниц на ваш сайт.

Таблицы и формы

Управление таблицами – одна из сильнейших сторон *Nvu*. Например, изменение размера таблицы на другой фиксированный размер выполняется простым перетаскиванием краев внутрь или наружу. Строки к существующей таблице добавляются почти так же, как в текстовых процессорах вроде *Microsoft Word* – просто нажмите Tab в самой нижней ячейке таблицы, и новая строка создастся автоматически.

Предусмотрено и управление формами. Интерфейс позволяет создавать формы для сайта, со всеми стандартными элементами, включая кнопки, поля ввода, текстовые области и так далее.

Поддержка CSS в *Nvu*, хотя и улучшена в некотором отношении по сравнению с *Mozilla Composer*, все еще неполная. Плюсом является то, что все аспекты представления, спроектированные в WYSIWYG-редакторе, оказывается, используют CSS: шрифты, размеры таблиц и многое другое – все это выполняется через каскадные таблицы стилей. А главный минус – нельзя проектировать разметку в несколько колонок, используя визуальный интерфейс. Насколько мы поняли, нет способа создать блоки DIV, или слои, как их называет *Dreamweaver*, которые вы смогли бы размещать, где вам угодно. Можно вызвать редактор CSS, чтобы сделать это самостоятельно, или ввести CSS-код вручную в редакторе исходного кода, но пользователю, не знакомому с CSS, для создания сайта с более чем одной колонкой придется прибегнуть к старой табличной верстке.

«Изменение размера таблицы — лишь вопрос перетаскивания краев внутрь или наружу.»

С этим связана неспособность данного web-редактора помещать графику в указанной точке документа. Место, где «приземлится» рисунок на странице, зависит от реализации вашего документа. Скажем, вы не сможете перетащить изображение в центр экрана, и ожидать, что редактор создаст нужный CSS-код для его соответственного позиционирования.

› Больше функций, чем можно упомнить, и несложный в работе интерфейс пользователя.

С другой стороны, следует подчеркнуть, что это не означает, будто в *Nvu* нет способа сделать подобное в визуальном режиме. Эти сбои связаны лишь с поддержкой CSS. Но вы по-прежнему можете задавать произвольное позиционирование рисунков и создавать сайты в несколько колонок, используя старый метод использования невидимых таблиц для разметки своего сайта.

Nvu – простой в использовании, бесплатный WYSIWYG-редактор со множеством функций, позволяющий создавать и сопровождать полнофункциональный web-сайт, наполненный формами. На данный момент, его основной недостаток – отсутствие поддержки серверных языков, таких как PHP и Perl, которые стали ключевыми во многих web-приложениях.

Вердикт

Nvu

Версия: 1.0
Сайт: www.nvu.com
Цена: бесплатно под MPL/LGPL/GPL

» *Приветливый визуальный редактор с отличным интерфейсом. Увы, ему недостает способности редактировать серверный код.*

Рейтинг 7/10

Quanta Plus

Приложение, основанное на *Qt*/KDE.

Quanta Plus не жалует WYSIWYG-подход; файлы проекта создаются в простом редакторе, снабженном панелями инструментов для вставки различных тегов. Панель инструментов Table включает в себя «мастер» быстрого создания таблиц, но можно и просто собирать их тег за тегом. При желании взглянуть, как выглядит проект, выберите в меню Preview или нажмите F6, и проект отобразится в *Konqueror*. Если предпочитаете *Netscape*, можете просмотреть свой проект и в нем, используя Shift+F5.

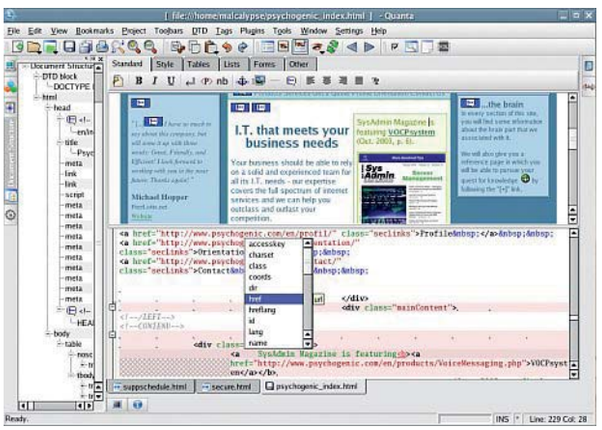
Наша любимая особенность *Quanta Plus* – интеграция с KDE; в частности, способность использовать KIO-slaves для доступа к удаленным серверам и файловым системам. Это предоставляет пользователям *Quanta* уникальную возможность сохранять свой контент непосредственно на web-сервере, запуская SSH, FTP, Samba или WebDAV. *Quanta* также имеет функцию проверки и подсветки синтаксиса, да не простую, а для 18 языков, включая PHP, Perl, SQL и HTML.

Подсветка синтаксиса по умолчанию вполне приемлема, но если у вас особые предпочтения, настройки можно поменять. Что касается проверки синтаксиса, *Quanta* строже большинства браузеров, и если ваш HTML прошел смотр в *Quanta*, то он годен к строевой.

Отличный «мастер» CSS позволит вам выбирать различные опции из выпадающего меню. Это намного проще, чем запоминать синтаксис каскадных стилей, и это быстрый способ создать правильные CSS. *Quanta* также включает генератор цветов, упрощающий их выбор для фона или текста. Он предельно прост в использовании: просто проведите мышью над палитрой, и соответствующее шестнадцатеричное значение определенного цвета появится в диалоговом окне.

Один из недочетов – проверка орфографии не умеет игнорировать общие HTML-теги (ну надо ли указывать, что 'html' – это не слово?), а также проверять орфографию выделенных фрагментов текста, например, абзацев.

«Генератор цветов упрощает их выбор для фона и текста.»



В интерфейсе *Quanta Plus* ваш web-сайт выглядит так же, как будет на сервере.

LINUX FORMAT Вердикт

Quanta Plus

Версия: 3.5.6
Сайт: http://quanta.kdewebdev.org
Цена: бесплатно под GPL

Фантастический редактор, предлагающий равные возможности как пользователям KDE, так и других сред, лишь с несколькими мелкими недочетами.

Рейтинг 8/10

Quanta Gold

Коммерческий вариант *Quanta Plus*.

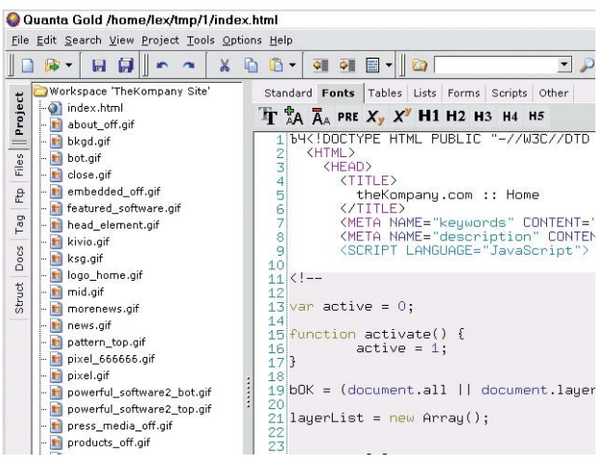
Интерфейсы *Quanta Plus* и *Quanta Gold* похожи. Есть основная панель инструментов, охватывающая базовый функционал «текстового процессора», такой как Save и Undo/Redo. Под основной панелью размещаются вкладки подменю, при выборе они отображают поднабор кнопок и идентичны в обоих проектах, за исключением Scripts, существующей только в *Quanta Gold*.

Текстовые редакторы в обоих приложениях весьма похожи, хотя (как и следовало ожидать) не одинаковы. *Quanta Gold* по умолчанию отображает слева номера строк и имеет серию присоединяемых «окон» для различных инструментов, часть из которых доступна лишь в *Quanta Gold*. В *Quanta Plus* такие «окна» для различных инструментов отсутствуют.

Едва начав работу с этими двумя приложениями, вы тут же увидите, какие они разные. Их интерфейсы, возможно, похожи, но ощущения от работы полностью различаются. *Quanta Plus* – «родное» приложение KDE 3, и выглядит и ощущается как типичное KDE-приложение: например, безупречно отображается в теме Liquid. А вот *Quanta Gold*, полностью построенная на библиотеках *Qt*, родным приложением KDE 3 не является и не использует KDE для отображения иконок, шрифтов, кнопок и т.д. Единственное исключение – *Quanta Gold* можно скачать в формате «shared-lib», позволяющем сглаживать шрифты в KDE.

Довольно странно, что функция проверки орфографии, имеющаяся, хотя и в ущербном виде, в открытом *Quanta Plus*, отсутствует в *Quanta Gold* [в этом нет ничего странного, особенно если учесть, что в *Quanta Plus* эту проверку выполняет *KSpell* – часть KDE, – прим. ред.]. Это раздражает многих пользователей, но им пообещали, что проверка орфографии скоро появится; и учитывая щедрую на обновления лицензию The Kompany, обещание весьма серьезное. Но я не назвал бы его «удачей»: может быть, это как раз одна из причин, что функции проверки орфографии в *Quanta Gold* до сих пор нет.

«В отличие от Plus, Quanta Gold строится на библиотеках Qt.»



Интерфейс коммерческого *Quanta Plus* очень простой, с подсветкой синтаксиса и просмотром структуры кода.

LINUX FORMAT Вердикт

Quanta Gold

Версия: 3.7
Сайт: www.thekompany.com
Цена: $39,95

Почти те же функции, что и в Quanta Plus, но без милого интерфейса KDE и не даром.

Рейтинг 6/10

Screem

Новичок в мире редакторов.

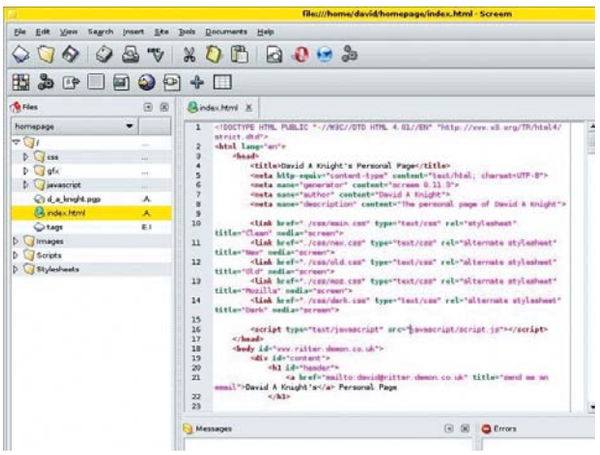
Когда вы приступаете к работе над сайтом с помощью *Screem*, идея заключается в создании проекта для связанных файлов. Вы определяете имя проекта, базовый локальный каталог, FTP-сервер для выгрузки и так далее, и после этого находите все файлы аккуратно уложенными в проект в одном месте. Впрочем, для редактирования страниц проект создавать не обязательно.

Написание и редактирование текста в *Screem* выполняется просто – хотя, возможно, и не столь просто, как могло бы быть. Как многие современные редакторы, он организован в виде меню и панелей инструментов на вкладках; версия ранняя, и панели инструментов пустоваты. К счастью, сгруппированный по функциям браузер тегов есть в левой панели. В списке категорий – Global Structure, Body Structure и Tables. Нам понравилась категория All, поскольку она заметно упрощает охоту на какой-нибудь неуловимый тег. Но следует отметить, что браузер тегов еще не закончен. Есть поддержка для HTML, PHP 3, RXML и SSI, и зарезервировано также место для JavaScript. Однако, при обширной поддержке CSS, в категориях они отсутствуют. А вот «мастер» таблиц работает безупречно.

В редактор включены браузер структуры (за неимением лучшего термина) и браузер свойств. Оба доступны вам в виде окна с вкладками в панели слева. Первый обеспечивает просмотр скелета тегов, использованных в вашем активном документе. Мы нашли бы его более полезным, если бы при этом можно было менять расположение тегов.

Screem вполне справляется с большей частью редактирования HTML. Будущие версии покажут, насколько он преуспеет в интеграции различных языков сценариев и разметки. На данный момент, некоторые части этого редактора напоминают стройплощадку и содержат несколько раздражающих ошибок, но его потенциал достаточен для многих создателей web-сайтов.

> «Панели инструментов пустоваты. К счастью, здесь есть браузер тегов.»



› *Screem* **может отпугнуть новичка, особенно сходством с Vi. На этом раннем этапе он многое обещает.**

LINUX FORMAT Вердикт

Screem

Версия: 0.16.1
Сайт: www.screem.org
Цена: бесплатно под GPL

» *Идеально удобный, но метит выше создания просто HTML-редактора; с интересом будем следить за его развитием.*

Рейтинг 4/10

OOo Web

Из текстового процессора в... во что?

По сути, HTML-редактор из *OpenOffice.org* – это просто расширение стандартного текстового процессора, Writer, без особых украшательств, привычных в настоящих web-редакторах. Единственный положительный момент – если вы уже используете *OpenOffice.org* для редактирования текстов, то легко шагнуть чуть подальше и приспособить его к использованию в web-дизайне.

Вы найдете *OpenOffice.org* Web в стандартном комплекте *OpenOffice.org*. Одна из серьезнейших проблем, с которой мы столкнулись в этом приложении – оно оказалось просто-напросто не инструментом web-дизайна. Ему недостает многих функций, используемых в наши дни в web-страницах, включая CSS. Генерируемый им код, откровенно говоря, ужасен, и даже если вы его почистите, при загрузке его обратно в *OOo Web* он опять превратится в мешанину. Ирония заключается в том, что этот код совместим с HTML 4.01, но если вы делаете что-то в *OOo Web*, не надейтесь, что потом это будет легко редактировать вручную.

Для начинающих, кому просто нужно что-нибудь базовое, *OOo Web* будет лучше, чем ничего, если вы уже используете *OOo* в своей офисной работе. Это ПО вполне интуитивно в использовании, так что создать нечто почти приличного вида будет легко, плюс следует очень простой терминологии: например, говорится «закладка», а не «ссылка». Превосходно для новых пользователей, но не обрадует тех, кто знает, что такое ссылка, и примется искать именно «ссылки».

Вероятно, единственное преимущество *OpenOffice.org* Web – его наличие по умолчанию в Linux-дистрибутивах. Он медленный, потребляет уйму памяти и не согласуется ни с какими стандартами популярных интерфейсов для Linux типа KDE или Gnome. Хорошо, если надо всего лишь наскоро создать web-страницу из существующего документа, чтобы раскидать его по сети, но разработка настоящего сайта в *OOo Web* – слишком трудная задача при слишком малой отдаче, чтобы тратить на нее усилия.

> «Разработка сайта в OOo Web — трудная задача с малой отдачей.»

› **Редактируете текстовый документ, как для печати, затем генерируете HTML. К сожалению, довольно уродливый...**

LINUX FORMAT Вердикт

OpenOffice.org Web

Версия: 2.0.4
Сайт: www.openoffice.org
Цена: бесплатно под LGPL

» *Если у вас уже есть OOo, почему бы не попробовать этот HTML-редактор; но есть свободные приложения и получше.*

Рейтинг 2/10

Web-редакторы

Вердикт

Quanta Plus 8/10

Было сложно назвать какой-то проект лучшим, просто потому, что многие имеют сравнимые возможности. Мы не ярые фанаты Gnome или KDE, так что это не повлияло на наш выбор; кроме того, и KDE-, и Gnome-приложения можно использовать в любом окружении. *Quanta Plus* и *Quanta Gold* выделялись предложением ряда фантастических функций для пользователей с любыми способностями; но *Quanta Plus* выполз вперед благодаря интеграции с KDE 3 и некоторым эргономическим реформам, сделавшим его приятнее в использовании.

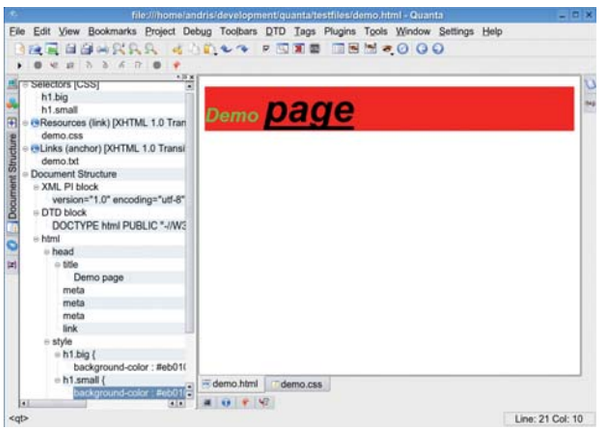
Нас сильно разочаровал уровень поддержки CSS почти во всех опробованных редакторах, хотя многие из них движутся в правильном направлении. Она представляет проблему для многих приложений этого жанра, поскольку новые web-стандарты и инструменты появляются каждую неделю, и остается только гадать, которые будут приняты наиболее широко.

Лошадки для скачки

Bluefish определенно целится на более опытного пользователя, предлагая подсветку синтаксиса и справку по функциям почти для всех популярных языков. Нам хотелось бы, чтобы *Bluefish* сосредоточил усилия именно на web-разделе своего репертуара. Как и многие открытые приложения, он движется в направлении, задаваемом пользователями и разработчиками. А вот *Nvu* все еще одной ногой вязнет в своем прошлом опыте с *Mozilla* и остро нуждается в улучшении функций для пользователей с более сложными потребностями; поддержка PHP и Perl будет первой в списке.

С XML, JavaScript и другими обработчиками на клиентской стороне, охотно используемыми в индустрии, почти любое из этих приложений будет только пытаться догнать коммерческие продукты типа *Dreamweaver*. Глянув на большинство открытых приложений, вы можете реально ожидать, что профессионал сумеет задействовать все их возможности. Но в целом, открытые web-редакторы не в силах противостоять другим доступным на рынке продуктам. Возможно, причина в отсутствии приложения-лидера, притягивающего энергию разработчиков.

Вместо него у нас есть семь приложений, покрывающих подавляющее большинство общих функций сайтостроения, но пока еще ни один из них не делает все очень хорошо. *Quanta* и *Bluefish* имеют перед собой огромные перспективы, как и *Nvu*, но лишь время покажет, какие приложения будут приняты пользователями. LXF



› В общем и целом, мы нашли *Quanta Plus* очень комфортной средой для работы.

Ваше мнение

Согласны ли вы с Дэвидом, что открытые web-редакторы – это бедные родственники проприетарных? Хотели бы вы видеть больше функций, помогающих управлять сайтом, вроде выгрузки, контроля версий и инструментов проверки целостности ссылок? Экономят ли визуальные редакторы ваше время? Или уж лучше простые текстовые редакторы, такие как *Vim* или *Nano*? Присылайте свои отзывы об этом Сравнении по адресу letters@linuxformat.ru.

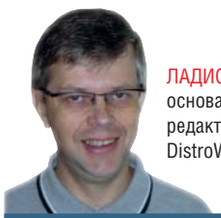
> «Нам бы хотелось, чтобы Bluefish сосредоточился на web-разделе своего репертуара.»

Таблица функций

	Bluefish	Nvu	OOo Web	Quanta Gold	Quanta Plus	Screem	SeaMonkey
Визуальный режим	☒	✓	✓	☒	☒	☒	✓
Таблицы	✓	✓	✓	✓	✓	✓	✓
Формы	✓	✓	☒	✓	✓	☒	☒
Фреймы	✓	✓	✓	✓	✓	✓	☒
XML	✓	✓	☒	✓	✓	☒	☒
CSS	✓	✓	☒	✓	✓	✓	✓
PHP	✓	☒	☒	✓	✓	✓	☒
Perl	✓	☒	☒	✓	✓	✓	☒
C	✓	☒	☒	☒	☒	☒	☒
FTP	✓	✓	☒	✓	✓	✓	☒
SSH	☒	☒	☒	☒	✓	☒	☒
WebDAV	☒	☒	☒	☒	✓	☒	☒
Проверка орфографии	✓	✓	✓	☒	✓	☒	✓
Управление проектами	✓	✓	☒	✓	✓	✓	☒

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux



ЛАДИСЛАВ БОДНАР
основатель, начальник, редактор и сотрудник DistroWatch.com.

PR-машина

Linspire вновь запустила ее. За последние несколько месяцев компания выпустила несколько пресс-релизов на различные темы, включая запуск сообщества дистрибутива Freespire, последнего мигранта в Ubuntu на волне «я-как-все» и стремление создать репозиторий пакетов для основных дистрибутивов. При этом Linspire продолжает отодвигать дату выхода Freespire 2.0, ранее обещанного к январю 2007.

Контраст между PR-акциями компании и ее неспособностью выдать обещанный продукт в срок настораживает. Если разработка Freespire 2.0 займет больше времени, как тогда они создадут Click-N-Run репозиторий программ для Freespire, Linspire, Ubuntu, Debian, Fedora и OpenSUSE к концу июня 2007?

Большие надежды

Поймите меня правильно. Репозиторий программ с постоянно обновляющимися пакетами, драйверами, проприетарными программами и лицензированными медиакодеками – и все это устанавливается одним щелчком мыши – прекрасная идея, которая упростит управление пакетами для многих дистрибутивов. Но я боюсь, что обещание Linspire создать к середине этого года решение, призванное устранить все проблемы по установке программ в Linux, нереально.

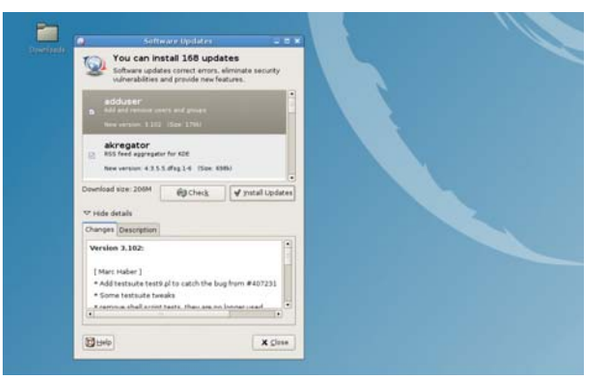
Написание пресс-релиза занимает несколько минут. Создание пакетов, отладка, тестирование и обновление программ различных версий для разных дистрибутивов требует тысячи и тысячи человеко-часов. Малость посидев над отладкой пакетов программ, ответственные за разработку PR-акций Linspire осознали бы это.

ladislav.bodnar@futurenet.co.uk

Большой Etch

Debian GNU/Linux 4.0 Крупнейший дистрибутив мира стал еще крупнее – запускайте ваш клиент BitTorrent прямо сейчас...

Стабильных выпусков Debian так мало и ждут их так долго, что выход каждой новой версии выглядит началом новой эры. Действительно, последний раз проект Debian выпускал стабильный релиз в июне 2005 г., а до него – в июле 2002. Ныне ожидаемый Debian GNU/Linux 4.0 'Etch' – это массивное обновление с огромным числом улучшений. Давайте пройдемся по новую версии и посмотрим, что нам приготовили.



› **Благодаря менеджеру обновлений *Ubuntu* и улучшениям в APT обновление безопасности и исправление ошибок в Debian проще некуда.**

Как и предыдущий релиз Debian, Etch побьет несколько рекордов. В нем будет около 20 300 программных пакетов, которые (для архитектуры i386) будут распространяться на 15 CD или трех DVD. Продукт будет доступен для 11 аппаратных архитектур, включая новинку – AMD64. Если вы решите заполучить все CD Debian 4.0 (включая диски с исходными текстами, занимающими куда больше места, чем двоичные пакеты), то вам придется загрузить более 300 CD-образов, что потребует около 200 ГБ трафика! Вы все еще на dial-up? Да не вопрос: приступайте прямо сейчас, и получите все уже к маю 2008.

Большинство тех, кто установит новый Debian, сделает это при помощи небольшого Netinst CD, который установит необходимый минимум с CD, а остальную часть дистрибутива загрузит из сети. Здесь пользователи, вероятно, впервые заметят серьезное нововведение Etch: утилиту установки «одной мышью». Хотя это не способ установки Debian по умолчанию (утилита вызывается командой *installgui*), но она присутствует на Netinst CD. Теперь любой, кто хоть раз устанавливал Fedora или OpenSUSE, сможет столь же легко установить Debian.

> «Как и все предыдущие релизы Debian, Etch бьет сразу несколько рекордов.»

Что касается программ, Etch использует ядро 2.6.18 и содержит *X.org 7.1*. Рабочий стол по умолчанию – Gnome 2.14, а стандартный браузер – *Epiphany 2.14*. Естественно, также доступны KDE 3.5.5, *Xfce 4.4* и большое число других оконных менеджеров. Присутствует и *Firefox 2.0*, но так как Mozilla Foundation возражала по поводу некоторых изменений в продукте, внесенных разработчиками Debian, то он включен под именем *IceWeasel*. По тем же причинам почтовый клиент *Thunderbird* назван *IceDove*. Наиболее популярный офисный пакет с открытым кодом *OpenOffice.org* поставляется в версии 2.0.4.

И это не все. Сервис Debian-volatile позволит вам легко обновлять программы, чья польза со временем падает, например, сканеры вирусов и спам-фильтры. Установлен SELinux, хотя и не включен по умолчанию, и разделы можно зашифровать в процессе установки системы (но не в графическом режиме установки). В качестве предпочтительного композитного менеджера поставляется Compiz, а также значительно улучшены поддержка и автоматическое обнаружение устройств.

Debian GNU/Linux 4.0 – отличный дистрибутив. Пользователи, которые не пожалеют времени на его исследование, обнаружат, что это не только крупнейшая из когда-либо созданных ОС, но также едва ли не лучше всех протестированный и стабильный дистрибутив Linux. Новый графический установщик, обновленный менеджер пакетов и красивый рабочий стол… любой ищущий основное и долговременное ОС-решение должен опробовать Etch, как только он выйдет.

www.debian.org

Красив и прост

PCLinuxOS 2007 Улучшенные скорость, дизайн и дрожание окон – массам.

Редкому пользователю Linux не нравится PCLinuxOS. Отпочковавшись от Mandriva Linux, дистрибутив завоевал популярность среди пользователей Mandriva, желающих, чтобы Linux работал сразу из коробки. Спустя месяцы разработки проект выпустил новую версию – PCLinuxOS 2007.

Этот релиз преподнес несколько сюрпризов. Во-первых, у него самое малое время загрузки, виденное нами у дистрибутивов Linux. На средней машине с Intel Pentium 4 загрузка занимает менее 30 секунд. Во-вторых, оформление рабочего стола переделано с использованием оригинальной графики и иконок. И в-третьих, для дрожащих окон, вращающихся кубов и других 3D-эффектов разработчики внедрили оба менеджера оформления: и *Compiz*, и *Beryl*.

Как и ранее, PCLinuxOS 2007 использует *APT* и *Synaptic*. Большинство стандартных расширений для браузеров, например, медиа-кодеки, *Macromedia Flash* и Java, устанавливаются по умолчанию, и все компоненты системы предусмотрены в последних, самых свежих версиях, включая KDE 3.5.6, *Firefox 2.0* и *OpenOffice.org 2.1*. Online-репозиторий дистрибутива теперь раздулся до 6 000 пакетов.

› Через центр управления легко настроить систему.

Хотя PCLinuxOS разрабатывался с прицелом на новичков в Linux, из отличной работы Билла Рейнольдса [Bill Reynolds] и его коллег, что собирали и отлаживали RPM-пакеты несколько лет, пользу сможет извлечь всякий. Не удивляйтесь, если эта жемчужина среди дистрибутивов отвоюет изрядную долю пользователей у других, более основательных, дистрибутивов Linux.

www.pclinuxos.com

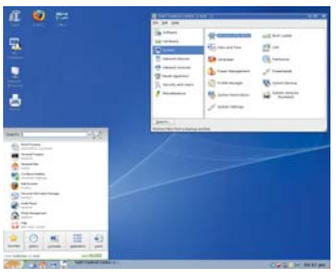
SUSE на все!

OpenSUSE 10.3 Улучшенная стабильность и безопасность от открытого детища Novell.

Первый альфа-релиз OpenSUSE 10.3 дал нам несколько подсказок о направлении развития популярного дистрибутива в следующие несколько месяцев.

В отличие от предыдущих релизов, версия 10.3 не имеет детального плана – только общую цель: выпустить финальный релиз OpenSUSE 10.3 в конце сентября 2007 г. (версия 10.2 была выпущена в декабре 2006 г.). Малое изменение версии с 10.2 на 10.3 говорит о том, что новый релиз не будет революционным. Вместо этого он сосредоточится на исправлении ошибок, обновлении пакетов и стабильности поставляемых утилит, особенно системы управления пакетами. Наиболее явные отличия – несколько обновленных пакетов: ядро Linux 2.6.20, KDE 3.5.6 и *OpenOffice.org 2.1.0*.

Тем не менее, значительные изменения произошли внутри. Как пояснил Андреас Егер [Andreas Jaeger], координатор релиза OpenSUSE, пакеты Gnome теперь устанавливаются в **/usr** вместо **/opt/gnome**, в русле других дистрибутивов. Более важно то, что весь дистрибутив собран с ключом fstack-protector, который снижает эффективность атак, использующих уязвимость типа «переполнение буфера».



› LUKS обеспечивает шифрование для жестких дисков и съемных устройств.

Как всегда, впереди длинная и трудная дорога компиляции и отладки. Но если все будет сделано, как обычно, в соответствии с высокими стандартами разработки OpenSUSE, то скоро мы сможем насладиться еще одним блестящим релизом. LXF

www.opensuse.org

Эволюция Debian

Итак, как мы пришли к Debian Etch? Дистрибутив Debian GNU/Linux официально запустил в августе 1993 Ян Мердок [Ian Murdock], студент колледжа. Ему так и не выпало удовольствия лично выпустить релиз: к моменту выхода первого стабильного Debian (версия 1.1, кодовое имя 'Buzz' ['Гул']) тремя годами позднее, он передал бразды правления Брюсу Пиренсу [Bruce Perens].

Первая стабильная версия Debian имела номер 1.1, а не 1.0 потому что первый набор CD содержал пакеты и код из нестабильного дерева и подлежал удалению несколькими днями позднее. Новый набор CD, пронумерованный как 1.1 во избежание путаницы, вышел спустя 6 месяцев.

Версия	Имя	Дата выхода	Версия ядра	Кол-во пакетов
Buzz	1.1	Июнь 1996	2.0.0	400
Rex	1.2	Декабрь 1996	2.0.27	540
Bo	1.3	Июнь 1997	2.0.29	890
Hamm	2.0	Июль 1998	2.0.34	1,330
Slink	2.1	Март 1999	2.0.36	2,250
Potato	2.2	Август 2000	2.2.16	2,980
Woody	3.0	Июль 2002	2.2.20	8,270
Sarge	3.1	Июнь 2005	2.4.27	16,850
Etch	4.0	Апрель 2007*	2.6.18	20,300
Lenny	4.1	2009*	?	?

*предварительно

Хит-парад дистрибутивов

10 самых посещаемых страниц на DistroWatch.com со 31 января по 2 марта 2007 г. (среднее число визитов в день)

Дистрибутив	Число визитов	
1 Ubuntu	3,105	‹›
2 SUSE	2,124	‹›
3 PCLinuxOS	1,794	‹›
4 Fedora	1,575	‹›
5 SimplyMepis	1,533	‹›
6 Debian GNU/Linux	1,162	‹›
7 Mandriva	1,020	↑
8 Sabayon Linux	864	↓
9 Mint	765	↑
10 Damn Small Linux	726	↓

» DistroWatch.com определяет популярность дистрибутивов, основываясь на числе визитов на страницу каждого дистрибутива. Хотя это не дает представление о действительном числе установок, но показывает, какие дистрибутивы более востребованы за определенный промежуток времени.

Убить спам

С жуткой эффективностью *SpamAssassin* блокирует письма, просачивающиеся через другие спам-фильтры. **Грэм Моррисон** расскажет, как этого достичь.

LINUX FORMAT На DVD
» SpamAssassin 3.1.8

Имя *SpamAssassin* [Спамоубийца, – прим. ред.] отлично подходит для голливудского фильма о фильтрации спама, даже неловко его тратить на какую-то там программу. Но так все же сделано, и приложение, известное, как *SpamAssassin* для многих из нас – лучшая возможность прикрыть свои почтовые ящики от злостных спамеров. За пять лет, прошедшие после регистрации *SpamAssassin* на *SourceForge*, он стал одним из наиболее уважаемых средств фильтрации почты – и среди бесплатных, и среди коммерческих.

SpamAssassin находится всего в паре «кликов» от вас, в менеджере пакетов вашего дистрибутива. Многие системы разбивают его на два инструмента: фоновый демон (*spamd*) и интерактивный клиент (*spamc*). Это гораздо эффективнее, чем запускать отдельную программу при каждой проверке почты. Даже среднестатистический пользователь может сканировать десятки писем за сессию, и использование демона устраняет накладные расходы на лишние запуски и остановы одного и того же процесса.

SpamAssassin больше похож на ниндзя, чем на викинга-берсерка: единожды настроенный и запущенный, он работает тишком. Сидя в фоне, *SpamAssassin* оценивает степень «спамности» каждого письма, сравнивая содержимое с примерами заведомого спама. Если эта степень превысит определенный порог, *SpamAssassin* делает что-то типа виртуального эквивалента рисованию красного креста на входной двери, оставляя свою метку в заголовках сообщения. В зависимости от конфигурации, он также может изменить тему сообщения или пнуть его прямо в **/dev/null**. Любой почтовый клиент сумеет фильтровать сообщения, выискивая письма с меткой *SpamAssassin*'а и сбрасывая таковые в папку карантина. В принципе, то же самое делают и другие основанные на сопоставлении фильтры спама, но отличительная черта *SpamAssassin* – чудовищная эффективность, отточенная за годы постижения древнего боевого искусства (как сказали бы голливудские сценаристы) байесовой фильтрации.

› **Вот чего вы ожидаете от *SpamAssassin*: письмо, помеченное как спам-сообщение, и предупреждение.**

Борьбу со спамом легко представить как сканирование текста сообщений в поисках стандартных фрагментов, присущих спаму. Эта технология известна как классификация текста. Слова типа «Viagra», «Ioan» и «Nigeria» отыскать легко, но простой поиск здесь не спасет. Спамеры давно научились обходить простые технологии фильтрации, фаршируя ключевые слова случайными символами и строками, точно так же, как какой-нибудь вирус модифицирует свой код ДНК, чтобы выжить.

Отношение «сигнал/шум»

Вот почему спам в наши дни напоминает скорее вывод «Энигмы», чем флайер случайного продавца. Наша любимая на сегодняшний день тема спама – «Я предумышленно убил вертекс» смысла не имеет, но фильтру не по зубам отличить его от «хорошего» письма. Байесовский фильтр спроектирован с обучаемостью постоянно меняющимся словам, используемым в спаме, и его секретное оружие – человек. Всякий раз, когда спам просачивается сквозь фильтр, пользователь «скармливает» его алгоритму обучения *SpamAssassin*, чтобы предотвратить повторную ошибку. Это ключ к успеху *SpamAssassin*: чем больше данных он получает, тем эффективнее становится; и спам удушен.

Вы можете протестировать байесов фильтр *SpamAssassin*'а из командной строки, не беря в голову демона или клиента. Сохраните письмо со спамом из вашего почтового клиента в текстовый файл и введите **spamassassin** с последующим именем файла, содержащего ваше сообщение. Вывод *SpamAssassin*'а встраивается в стандартный формат электронного сообщения и промчится мимо вас в командной строке, если вы не перенаправите его в отдельный файл, используя **>new_file.txt**.

SpamAssassin добавляет несколько строк в заголовок всех просканированных писем, будь то спам или нет, неважно. Заголовки сообщений (как правило, скрываемые клиентами электронной почты) содержат данные для передачи сообщения от отправителя получателю, а также путь, пройденный письмом за путешествие до вашего почтового ящика. Как минимум, *SpamAssassin* добавляет три заголовка: X-Spam-

Checker-Version, X-Spam-Level и X-Spam-Status. Поле Level используется для отображения степени сходства письма со спамом: набранный балл отображается символами «звездочка». Если их больше пяти, письмо, вероятно, является спамом. По данным поля Level почтовые клиенты типа *KMail* выводят кнопку предупреждения достаточно устрашающего вида.

Результаты экзамена

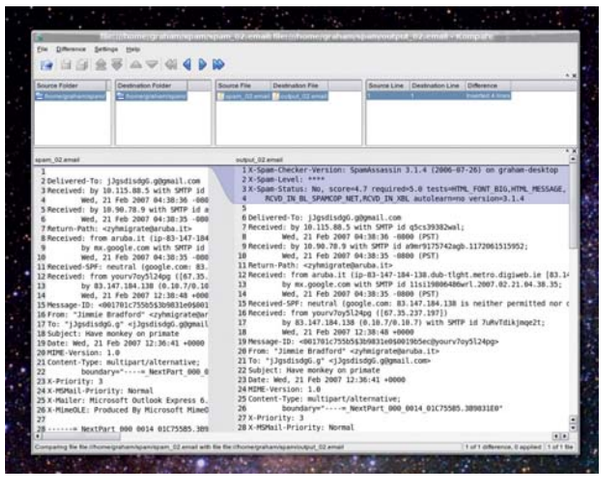
Но пища для размышления может быть найдена в последнем поле, Status. Оно подытоживает обработку, выполненную над сообщением, и то, как *SpamAssassin* определил похожесть сообщения на спам. Ниже показан статус спам-сообщения, которое мы пропустили через *SpamAssassin* с настройками по умолчанию:

```
X-Spam-Status: No, score=4.7 required=5.0 tests=HTML_FONT_
BIG,HTML_MESSAGE,
RCVD_IN_BL_SPAMCOP_NET,RCVD_IN_XBL autolearn=no version=3.14
```

Итоговый балл этого письма – 4.7, он не дотягивает до порога 5.0, определенного в поле **required**. Тесты, использованные для определения балла, перечислены в поле **tests**. В нашем случае на письмо отреагировали лишь четыре теста. Первые два обнаружили, что оно является HTML-сообщением с крупными шрифтами (**HTML_FONT_BIG и HTML_MESSAGE**). Последние два теста сверяют IP-адрес в заголовке сообщения с черными списками известных преступников, размещенных на некоторых серверах в сети. Несмотря на эти тесты, *SpamAssassin* не смог набрать достаточно предупреждений, чтобы пометить это сообщение как спам. Одно из решений – отредактировать конфигурационный файл в **~/.SpamAssassin/user_prefs** и понизить параметр **required_score** с 5.0 до 4.5, но в результате *SpamAssassin* с большей вероятностью будет ошибочно помечать нормальные письма как спам.

Лучше научить *SpamAssassin* распознавать этот тип сообщений как спам, указав ему на эту ошибку. Используя команду sa-learn, вы можете либо обучить *SpamAssassin* распознавать спам, который был пропущен, либо повторно рассмотреть сообщения, которые были восприняты как спам (они помечаются как 'ham'). Чтобы обучить *SpamAssassin*, наберите либо **sa-learn --spam**, либо **sa-learn --ham**, с последующим именем файла, содержащего сообщение. Чем на большем числе писем вы сможете провести обучение, тем лучше ваша защита.

Именно обучаемость делает *SpamAssassin* столь мощным. Потратив время и потренировав его на письмах из вашего почтового ящика, вы сможете ожидать, что спам вас больше не побеспокоит. Но лучшее, что вы можете сделать для противостояния спаму – никогда не отвечать ни на какие спам-сообщения. Единственное, что в глазах спамера оправдывает его грязный бизнес – это одно письмо из миллиона, приносящее ему некоторые деньги. Вы удивитесь, но некто даже взял да и щелкнул на «ответить», получив попрошайническое письмо якобы от имени вдовы Ясира Арафата. LXF

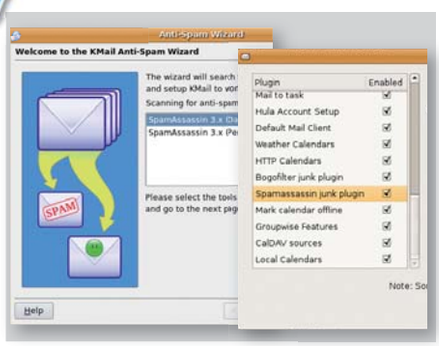


Инструмент KDE Kompare покажет, что именно *SpamAssassin* добавил в заголовки ваших электронных сообщений.

Советы по SpamAssassin

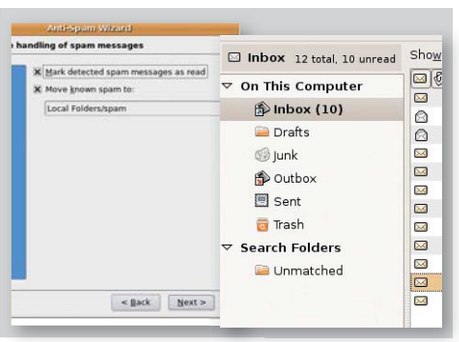
- Сохраняйте свой спам! *SpamAssassin* учится на своих ошибках. Это означает, что вам нужно использовать ваши спам-сообщения для обучения фильтра.
- Не просто переносите ошибочно отфильтрованные сообщения назад в свой почтовый ящик. Позаботьтесь, чтобы *SpamAssassin* больше не сделал такую же ошибку: пропустите 'ham' через утилиту *sa-learn*.
- Проверяйте сообщения у источника. Если у вас запущен почтовый сервер, установите простой скрипт для проверки входящей почты. Если вы платите кому-то за обслуживание вашей почты, попросите его установить *SpamAssassin*.
- Интегрируйте *SpamAssassin* в свой почтовый клиент. Даже без встроенной поддержки вы все же можете пропускать почту через *SpamAssassin*, используя фильтр.
- Если ваш клиент не интегрируется со *SpamAssassin*, сортируйте спам и ошибочно помеченные письма по своим собственным папкам. Вы можете затем обучать *SpamAssassin* из командной строки.
- Запускайте *SpamAssassin* как демон. Команда *spamassassin* удобна, но интенсивно нагружает процессор. Используйте вместо нее *spamd* и *spamc*.

Шаг за шагом: Настройки для KMail или Evolution



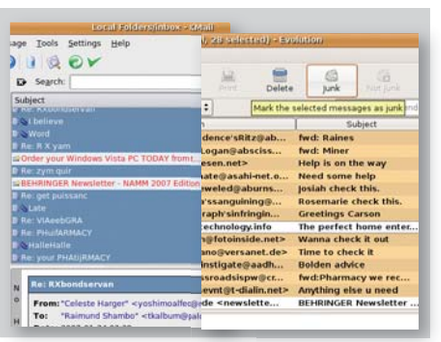
1 Установите SpamAssassin

Последние версии *KMail* (слева) и *Evolution* (справа) имеют встроенную поддержку *SpamAssassin*. *KMail* использует Мастер (Anti-Spam Wizard), а *Evolution* – расширение. Для работы обоих нужно убедиться, что установлен *SpamAssassin*, а также демон *spamd* и клиент *spamc*.



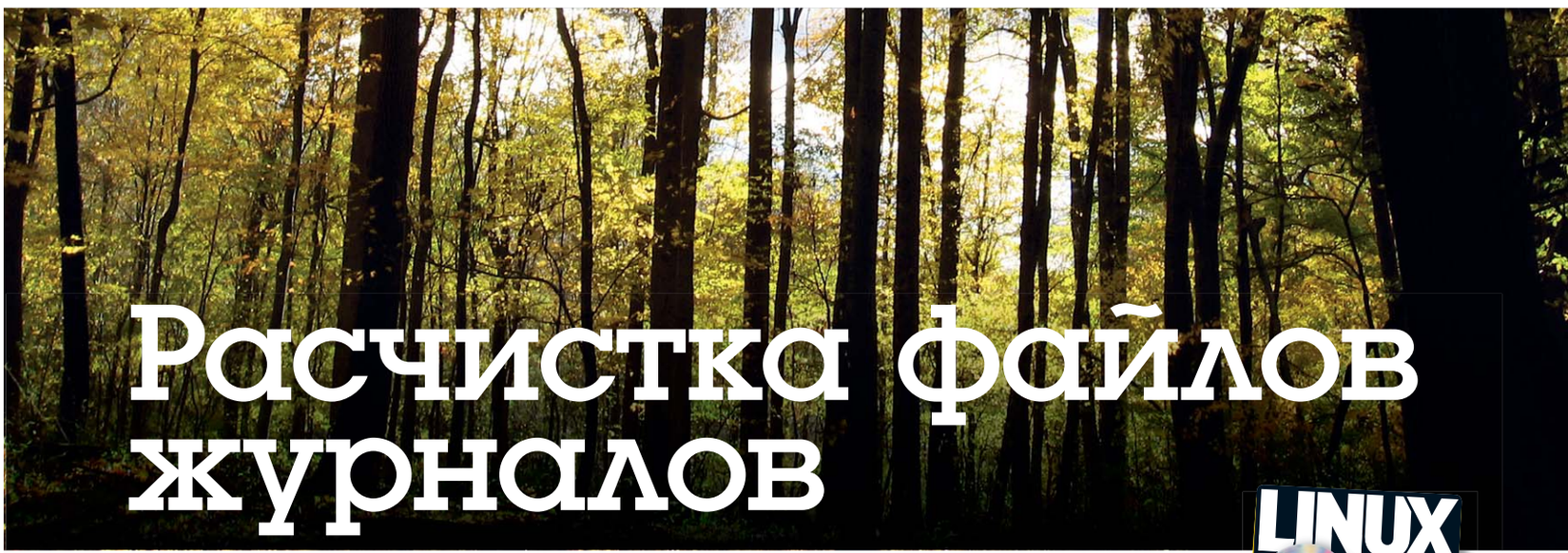
2 Задайте пункт назначения

В *KMail* создайте папку «spam» и укажите ее как место назначения в Мастере. *Evolution* создает папку «junk» и помещает спам в нее автоматически. Не позволяйте почтовому клиенту удалять сообщения до ручной проверки: *SpamAssassin* поначалу может допускать некоторые ошибки.



3 Помогите ему учиться!

В *KMail* выделите любой спам, который не определился автоматически, и щелкните по круглой зеленой иконке. В *Evolution* щелкните по иконке Junk. Есть и иконки для пометки сообщения как ошибочного причисленного к спаму, их *SpamAssassin* тоже использует для обучения.



Расчистка файлов журналов

За деревьями не видите леса? Урежьте файлы журналов по разумным правилам, и вашими данными станет проще управлять. Д-р **Крис Браун** вам все объяснит.

В вашем ведении – Linux-системы, критичные для вашего бизнеса? Web-сервер, сервер баз данных или почтовый сервер, и, возможно, пограничные укрепления типа межсетевого экрана? Знайте, что мониторинг состояния этих систем жизненно важен, и файлы журналов – ваше первое прибежище: они могут подсказать, что настроено не так, предупредить о попытках взлома или просто подтвердить, что все идет как надо.

В прошлый раз мы разбирались, как ведется запись в файлы журналов. Сейчас мы бегло просмотрим содержимое некоторых из них и поймем, что там находится; затем изучим инструменты, помогающие анализировать журналы и управлять ими.

Сообщения в журналах делятся на две категории: маленькие, которые можно читать по строкам, и большие, которые невозможно охватить без дополнительной обработки. Давайте рассмотрим несколько коротких записей.

```
e1000: 0000:04:03.0: e1000_probe: (PCI:66MHz:32-bit)
    00:0e:0c:4a:42:4e
e1000: eth0: e1000_probe: Intel(R) PRO/1000 Network Connection
e1000: eth0: e1000_watchdog_task: NIC Link is Up 10 Mbps Full
    Duplex
```

Экспонат A взят из кольцевого буфера ядра, который мы обсуждали в прошлый раз. Его строки показывают, что модуль ядра обнаружил и инициализировал сетевой интерфейс – важное подспорье, если вы добавили новое оборудование и пытаетесь проверить, распознает ли его Linux.

```
Feb 6 15:11:56 shrek named: zone localdomain/IN: loaded serial
    42
Feb 6 15:11:56 shrek named: zone localhost/IN: loaded serial 42
Feb 6 15:11:56 shrek named: named.local:9: unknown RR type
    'PT'
```

Экспонат B, взятый из журнала общего назначения, **/var/log/messages**, показывает, что DNS-сервер не запустился из-за синтаксической ошибки в одном из файлов зон. **Feb 6 15:11:56** – дата события; **shrek** – имя компьютера. Далее, **named** – имя сервиса, рапортующего о проблеме, квалифицированной как **loaded serial 42**. Третья строка сообщает подробности, показывая номер строки и имя файла, где произошла ошибка.

Следующие три примера представляют собой сообщения, которые в большом количестве попадаются даже на здоровых машинах.

```
Feb 6 15:08:06 shrek su: pam_unix(su:session):
    Session opened for user root by pete (uid=571)
```

Экспонат C, из **/var/log/secure**, показывает, что обычный пользователь получил права root с помощью su. На данной конкретной машине запрещен прямой вход root'а, и при попыке получить полномочия root пользователи обязаны раскрыть свою личность (а значит, оставить контрольный след).

```
Feb 2 18:54:16 shrek sshd[4244]:
Failed password for root from 125.246.84.5 port 39409 ssh2
```

Экспонат D демонстрирует неудачную попытку захода root по SSH. Увидев одно такое сообщение, вы подумаете, что, наверное, косорукий пользователь ошибся при вводе пароля, но осознав, что их 10 000, и что уже 12 часов подряд они повторяются каждые пять секунд...

```
194.105.66.18 - - [05/Feb/2007:08:31:38 +0000]
"GET /media/images/premiumskin07d/bg.gif HTTP/1.1" 404 320
```

Наконец, экспонат E – строка из журнала доступа к *Apache*. Это пример типового формата журнала, о котором говорилось в **LXF91**. Конечно, web-сервер и предназначен для обращений, поэтому появление новых строк само по себе не сюрприз; однако именно эта строка интересна тем, что говорит об ошибке **404** (файл не найден), подавая, возможно, сигнал о битой ссылке на сайте. Кстати, если вы настроите собственный формат журнала, *Apache* запишет в него более детальную информацию, включая содержимое любых полей заголовка HTTP-ответа. На самом деле, существует довольно стандартное расширение, называемое комбинированным форматом журнала, который понимают многие инструменты анализа журналов (более подробно об этом читайте на http://httpd.Apache.org/docs/1.3/logs.html#combined).

В принципе, никому не охота строка за строкой разбирать записи типа C, D и E. Хорошо бы кто-то проанализировал их за вас и выдал вам отчет, например: 'pete хотел стать root-пользователем и использовал команду *su* 26 раз,'.

Стать супер-чистильщиком

Без вмешательства размер журнала выйдет за пределы разумного, а большая часть его данных устареет до полной бесполезности. Чтобы оставаться на гребне волны, отсекайте части журналов и посылайте старые журналы в архив. Для этого в дистрибутивах Linux используется инструмент *logrotate*. Он занимается ротацией журналов, то есть периодически разделяет их на осмысленные части и, со временем, удаляет старые. Обычно *logrotate* запускается раз в день как задача *Cron*. У *logrotate* есть собственный файл настройки, определяющий, какими он

управляет журналами – укажите, как часто производить их смену (раз в день, неделю или месяц, или когда они достигнут определенного размера, который также можно задать). Старые журналы можно сжать и отправить по почте заинтересованным пользователям.

Главный файл настройки для *logrotate* находится в **/etc/logrotate.conf** – взглянем на него. Он определяет некоторые параметры по умолчанию и использует директиву **include** для включения файлов настройки отдельных сервисов из каталога **/etc/logrotate.d**. Исключая строки комментариев, на тестовой установке Fedora Core 5 **logrotate.conf** выглядит следующим образом (для удобства ссылок здесь вставлены номера строк):

```
1. weekly
2. rotate 4
3. create
4. include /etc/logrotate.d
5. # no packages own wtmp -- we'll rotate them here
6. /var/log/wtmp {
7. monthly
8. create 0664 root utmp
9. rotate 1
10. }
```

В строке 1 мы велим *logrotate* обновлять журналы еженедельно. Строка 2 говорит, что хранятся последние четыре файла журналов; в контексте данного примера это означает журналы за последние четыре недели. Когда происходит ротация, текущий журнал (например, **/var/log/messages**) переименуется в **messages1**; файл **messages1** переименуется в **messages2**, и так далее. Строка 3 сообщает, что после смены журналов создадутся новые журналы. Можно настроить работу *logrotate* для каждого журнала в отдельности: например, журнал **/var/log/messages** менять каждый день и оставлять семь его поколений, а журнал **/var/log/mail.log** – каждую неделю, и оставлять последние четыре журнала.

Строка 4 говорит *logrotate*, чтобы он читал все файлы настройки из **/etc/logrotate.d**. В этом каталоге находится набор файлов, каждый из которых приписан к определенному сервису (или, более точно, к определенному журналу). Такое разбиение настроек позволяет упростить конфигурацию: по мере установки программ просто добавляются новые файлы настройки в каталог **/etc/logrotate.d**. Однако вполне возможно добавлять настройку определенных сервисов (или проводить всю настройку) прямо в **logrotate.conf**, именно такой пример мы видим в строках 5–10, с настройкой замены файла **wtmp**.

Примером отдельного файла настройки служит **/etc/logrotate.d/ppp** (также из Fedora Core 5). Данный файл управляет настройкой смены журнала, создаваемого демоном *ppp*, который управляет телефонными соединениями.

```
1. /var/log/ppp/connect-errors {
2. missingok
3. compress
4. notifempty
5. daily
6. rotate 5
7. create 0600 root root
8. }
```

Пройдемся по нему. Строка 1 описывает, к какому журналу применяется вся запись. В строке 2 *logrotate* велено не жаловаться, если журнала нет. Строка 3 требует, чтобы старые (уже смененные) журналы сжимались – в этом примере они примут вид **connect-errors.1.gz**. В строках 4 и 5 администратор указал, чтобы смена журналов не происходила, если журнал пуст, в противном случае журналы сменяются ежедневно (заметим, что эти настройки имеют преимущество над настройками по умолчанию в **logwatch.conf**). Понятно, что при таком раскладе необходимо запускать *logrotate* хотя бы раз в день.

Оставшиеся настройки должны быть вам теперь ясны: строка 6 говорит о сохранении пяти поколений журналов (подавляя настройки по умолчанию), а строка 7 (также подавляет настройки по умолчанию) говорит, что после смены журнал должен быть создан с заданными правами доступа. Существует еще несколько директив *logrotate*, указывающих, как работать с журналом; посмотрите man-страницы *logrotate*, чтобы узнать подробности.

Результат смены журналов можно увидеть, просмотрев содержимое каталога **/var/log**. Ниже приводится пример четырех смен журнала **maillog**:

```
$ ls -l /var/log/maillog*
-rw------- 1 root root 4112 Jan 19 04:02 /var/log/maillog
-rw------- 1 root root 6584 Jan 14 04:02 /var/log/maillog.1
-rw------- 1 root root 5771 Jan 7 04:02 /var/log/maillog.2
-rw------- 1 root root 7794 Dec 31 04:02 /var/log/maillog.3
-rw------- 1 root root 5561 Dec 24 04:02 /var/log/maillog.4
```

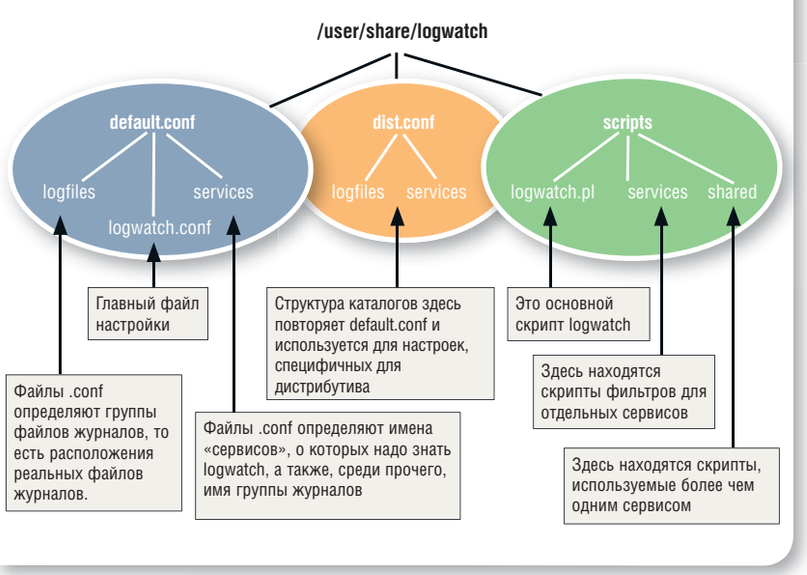
Вы можете видеть, что все эти файлы были созданы в 4:02 утра (не потому, что нервный сисадмин сорвался с койки и запустил *logrotate*, а потому, что запуск *logrotate* прописан как задача в *Cron*), но смена их происходит только раз в неделю. Так как мы держим только четыре последних поколения, то почтовые журналы, записанные до 17 декабря (когда был начат **maillog.4**), удалены. В этом примере файлы не сжимаются.

Наблюдаем за журналами

Хотя *logrotate* занимается проблемой разделения журналов на части требуемого размера, никакого анализа содержимого здесь не делается. Существует ряд открытых приложений (и несколько проприетарных), способных нам в этом помочь. Один из широко используемых инструментов – *logwatch*, простая, но полезная программа, выполняющая всю черновую работу по чтению и анализу журналов, так что вместо просмотра десятков тысяч строк текста журнала *Apache* вы получите отчеты с графиками использования сайта за день, час, по URL и так далее.

Logwatch доступна в большинстве дистрибутивов Linux. Как и *logrotate*, она обычно запускается ежедневно как задача *Cron*, и обычно настроена на отправку своего вывода по почте пользователю root; и если вы войдете как root на свой позабытый сервер и почитаете почту, то наверняка увидите длинный список ежедневных депеш от *logwatch* (и самую малость других писем). Сердцевина *logwatch* – Perl-скрипт **/usr/share/logwatch/scripts/logwatch.pl**. Но к нему еще поставляется малодоступная для восприятия иерархия файлов настройки и целый набор фильтров, вырезающих информацию из журналов о сервисах, которые вас не интересуют. В Fedora Core 5 *logwatch* запускается раз в день благодаря символьной ссылке на главный скрипт **logwatch** в каталоге **/etc/cron.daily**. »

Структура каталогов logwatch



» Большинство пользователей спокойно оставят для *logwatch* настройки по умолчанию, но когда вы освоитесь в запутанном лабиринте настроек и скриптов, то сможете много чего понаделать: например, написать собственные скрипты фильтров, если вы обладаете нужными навыками программирования – скорее всего, в Perl или PHP. *Диаграмма Структура Каталогов Logwatch* на предыдущей странице показывает основную структуру *logwatch*, корнем которой является **/usr/share/logwatch**. Крайний левый эллипс на рисунке (каталог **/usr/share/logwatch/default.conf**) содержит файлы, определяющие базовую настройку. Файлы в этом каталоге одинаковы для всех установок *logwatch*. Эллипс посередине (каталог **dist.conf**) содержит подкаталоги, аналогичные **default.conf**. В них содержатся настройки, специфичные для дистрибутива.

На самом деле, есть и третий каталог, **/etc/logwatch/conf** (на диаграмме не показан), дублирующий каталог **default.conf**: в нем содержится настройка под конкретную машину. Идея заключается в том, что вы должны не портить настройки по умолчанию, а использовать настройки для дистрибутива и машины для переопределения или расширения настроек (на нашей тестовой системе с Fedora настройки для дистрибутива и машины пусты). Так поступают не все дистрибутивы: Red Hat Enterprise Linux, например, имеет только один такой каталог, корень которого находится в **/etc/log.d**.

Центральным для настройки *logwatch* является понятие «сервис». Мы чувствуем себя обязанными заключить это слово в кавычки, потому что не имеем в виду конкретный сетевой сервис типа DNS или FTP, которые представлены особыми демонами в Linux. На языке *logwatch*, сервис – это просто некий источник журналируемой информации, идентифицируемый по имени. Все понятно?

Вы, вероятно, уже изнемогли от обилия сервисов; вернемся к любезным сердцу файлам настройки, точнее, к файлу настройки *logwatch* **/usr/share/logwatch/default.conf/logwatch.conf**. Вот он, лишенный комментариев (но с добавленными номерами строк):

```
1. LogDir = /var/log
2. TmpDir = /var/cache/logwatch
3. MailTo = root
4. Print = No
5. Range = yesterday
6. Detail = Low
7. Service = All
8. Service = "-zz-network"
9. Mailer = /bin/mail
```

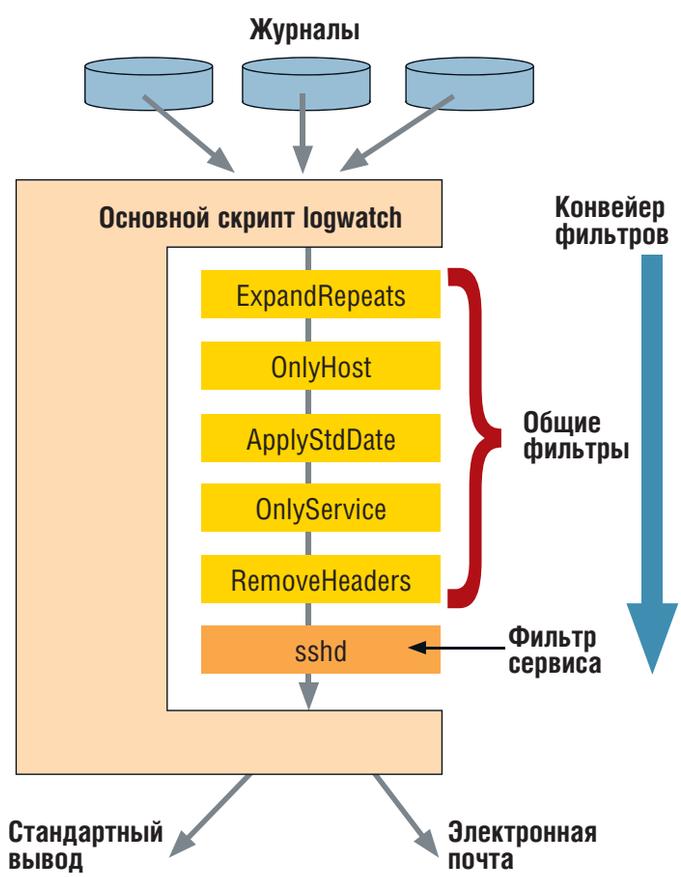
Строка 1 говорит *logwatch*, где искать файлы журналов. Мы займемся этим через минуту, когда увидим, как определять группы журналов. Строки 3 и 4 объясняют, что вывод *logwatch* следует отправлять root по почте, а не выводить на STDOUT. Строка 5 определяет отметку времени, с которого начинается обработка записей в журнале (допустимые значения – **yesterday**, **today** и **all**), а строка 6 определяет уровень детализации получаемого отчета – имеются значения **Low**, **Med**, **High**, а как их интерпретировать, это уже дело фильтров данного сервиса. Если вы намерены обработать все сервисы, то установите в строке 7 значение **All**, как в нашем примере, но можно предусмотреть исключения, как в строке 8, которая явно запрещает сервис *zz-network*. Строка 9 указывает положение почтового агента.

Прежде чем двигаться дальше, стоит упомянуть, что большинство настроек можно переопределить из командной строки при вызове *logwatch*. Например, команда

```
# logwatch --detail High --service cron --print
```

обработает только сервис *Cron*, с высоким уровнем детализации, и результат отправится на стандартный вывод, а не по почте. Наберите **man logwatch** или **logwatch --help**, чтобы узнать подробности о дополнительных опциях командной строки.

Цепочка фильтров для данных



Объединяемся в группы

Далее в нашем туре по файлам настройки мы остановимся на определении сервисов, расположенных в **default.conf/services**. Файлы в этом каталоге определяют сервисы, о которых следует знать *logwatch*. Например, вот файл **sshd.conf** (комментарии опущены):

```
1. Title = "SSHD"
2. LogFile = secure
3. LogFile = messages
4. *OnlyService = sshd
5. *RemoveHeaders
```

Строка 1 определяет название сервиса. Любой вывод, генерируемый фильтром для *sshd*, будет обрамлен так:

```
-------- SSHD Begin ---------
--------- SSHD End ----------
```

В строках 2 и 3, **secure** и **messages** – это имена пары групп журналов. Группы журналов просто служат для назначения настроек, общих для нескольких определений сервисов. Группы определяются файлами в **default.conf/logfiles**; в нашем случае это **secure.conf** и **messages.conf**. Мы рассмотрим их далее, но попросту говоря, каждая группа журналов определяет один или более файлов, которые будут рассматриваться как источник журналируемых сообщений. Строка 4 определяет фильтр **OnlyService**, запускаемый с параметром **sshd**. Это один из разделяемых скриптов в каталоге **cripts/shared**; цель данного фильтра – отбор строк, начинающихся с sshd. Другой общий фильтр находится в строке 5: он удаляет «мусор» в начале каждого syslog-сообщения. Мусором мы полагаем отметку времени, имя узла и имя владельца – администратору-исследователю они ни к чему.

Время взглянуть на одно из определений группы журналов. Возьмем, к примеру, файл **secure.conf**. Вот он:

```
1. LogFile = secure
2. LogFile = authlog
3. Archive = secure.*
4. Archive = secure.*.gz
5. Archive = archiv/secure.*
```

```
6. Archive = archiv/secure.*.gz
7. Archive = authlog.*
8. Archive = authlog.*.gz
9. *ExpandRepeats
10. *OnlyHost
11. *ApplyStdDate
```

Строки 1 и 2 определяют имена настоящих файлов журналов. Здесь вы можете использовать абсолютные имена, а если нет, то имена берутся относительно каталога **LogDir**, определенного в основном файле настройки и указывающего на **/var/log**. В данном примере содержимое **/var/log/secure** и **/var/log/authlog** передается в цепочку фильтров для этого сервиса.

Строки 3–8 определяют местоположение архивов журналов; они используются, только если *logwatch* запускается с опцией **--archives** или если строка **Archives = yes** определена в главном файле настройки. Строки 9, 10 и 11 определяют последующие три фильтра (также в каталоге **scripts/shared**), которые будут помещены в начале цепочки фильтров. Фильтр ExpandRepeats был изначально спроектирован для надставки последнего сообщения, которое повторяется N раз, сообщениями syslog (как вы помните из прошлого урока), но авторы, видимо, передумали, и теперь фильтр просто подавляет эти сообщения. OnlyHosts фильтрует сообщения в соответствии с узлом, от которого они пришли – надеюсь, вы помните, что syslog может принимать сообщения от других узлов сети. Этот фильтр используется совместно с командной опцией **--splithosts**, она заставляет *logwatch* создать отдельный отчет для каждого узла, обнаруженного в исходных журналах. Наконец, фильтр ApplyStdDate отклоняет те строки, временные отметки которых не попадают на обрабатываемый день.

Все эти настройки «сшивают» вместе цепочку обработки фильтров, которые работают под управлением скрипта *logwatch.pl*, как показано во врезке «*Цепочка фильтров для данных журнала*», слева. Показанные здесь общие фильтры располагаются в **/usr/share/logwatch/scripts/shared**; сервис-ориентированные скрипты – в **/usr/share/logwatch/scripts/services**. Именно сервис-ориентированные скрипты генерируют вывод.

Пропажа файлов

Если хотите увидеть *logwatch* в действии, можете запустить ее прямо из командной строки, благодаря символьной ссылке в **/usr/sbin**. Понадобится также определить опцию **--print**, чтобы увидеть вывод, так как по умолчанию вывод отправляется на почту root. Вот простой запуск:

```
########## Logwatch 7.2.1 (01/18/06) ############
Processing Initiated: Thu Feb 1 04:02:02 2007
Date Range Processed: yesterday
( 2007-Jan-31 )
Period is day.
Detail Level of Output: 0
Type of Output: unformatted
Logfiles for Host: www.example.com
################################################
-------------------- httpd Begin --------------------
Requests with error response codes
400 Bad Request
/: 1 Time(s)
404 Not Found
/media/images/premiumskin07d/bg.gif: 51 Time(s)
/favicon.ico: 3 Time(s)
--------------------- httpd End ---------------------
------------------- pam_unix Begin -----------------
sshd:
Authentication Failures:
unknown (61.19.233.102): 27 Time(s)
admin (61.19.233.102): 3 Time(s)
admin (61.136.58.249): 1 Time(s)
root (61.136.58.249): 1 Time(s)
unknown (61.136.58.249): 1 Time(s)
Invalid Users:
Unknown Account: 28 Time(s)
su:
Authentication Failures:
pete(571) -> root: 1 Time(s)
Sessions Opened:
pete(uid=571) -> root: 3 Time(s)
```

Что мы можем извлечь из этого вывода? Похоже, на нашем web-сайте не хватает файла. Судя по названию, это нечто косметическое. Во-вторых, было некоторое число неудачных попыток зайти в систему по sshd (*sshd* и *Apache* – единственные сервисы, включенные на нашей машине). В-третьих, пользователь Pete заходил как root три раза (и случайно ошибся один раз; с кем не бывает, Pete). Так как прямой доступ root на этой машине запрещен, все попытки (удачные или неудачные) будут зарегистрированы, как показано здесь. Короче, ничего примечательного нет. Просто демоны делают свою рутинную работу.

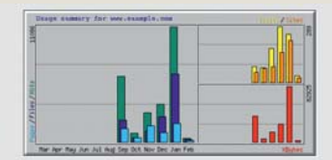
Logwatch – не единственная рыбка в этом океане. Пользователи Debian возможно, напомнят нам о *logcheck*, которая выкидывает «нормальные» строки из журналов, пользуясь регулярными выражениями, а затем отправляет оставшиеся «аномальные» строки системному администратору. И, конечно, если у вас есть навыки Perl или PHP, вы сможете и сами сделать нечто подобное.

Ну, вот и все. Журналы. Мы вас предупреждали, что захватывающего в них мало! Но в дождливый день, если вы просмотрели газету, прочли Linux Format от корки до корки и вам нужен предлог, чтобы не мыть кошку, вы всегда можете скрыться в своем кабинете и почитать файлы журналов. LXF

Анализ web

Когда дело доходит до анализа журналов web-сервера, то вы можете использовать один из замечательных инструментов вроде Google Analytics и Clicktracks. Они помогут отделу маркетинга установить, как люди путешествуют по вашему сайту и откуда они туда попадают.

Открытый инструмент *The Webalizer* (www.mrunix.net/webalizer) – более традиционный: он считывает журналы, написанные *Apache*. Зато вы получаете статистику использования в HTML-формате, которую можно просмотреть в браузере. Можно получить статистику по годам, месяцам, дням и часам, а также отобразить статистику по сайту, URL, местам, откуда пришел пользователь, строке User Agent (т.е. браузеру), имени пользователя, строке поиска, странице входа/выхода и стране. Снимок экрана *справа вверху* является примером вывода; каждая подчеркнутая ссылка в таблице ведет к более подробной статистике для этого месяца, включая графики использования по дням и часам, а также список популярных адресов URL.



Summary by Month

Month	Daily Avg				Monthly Totals					
	Hits	Files	Pages	Visits	Sites	KBytes	Visits	Pages	Files	Hits
Feb 2007	56	38	22	5	21	2445	33	137	229	337
Jan 2007	357	210	57	8	216	82925	250	1785	6535	11086
Dec 2006	117	78	29	9	154	26868	289	918	2427	3649
Nov 2006	97	36	53	6	82	14386	174	1554	1048	2832
Oct 2006	26	13	12	2	73	4770	76	399	412	823
Sep 2006	330	110	68	4	51	38413	82	1308	2092	6287
Totals						169807	904	6101	12743	25014

Практично, что The *Webalizer* поддерживает собственный коллектор статистики деятельности сайта, так что он может предоставить данные, которые уходят в прошлое намного дальше, чем способен обеспечить движок смены файлов журналов. Он также поддерживает журналы передач по FTP с сервера wu-ftpd, и при необходимости сам разархивирует сжатые журналы.

ОБ UBUNTU

«Может ли Debian чему-то поучиться у Ubuntu? Конечно.»

Photography: Jena Cumbo

Наберите M, чтобы поговорить с Мердоком

Сегодня он трудится, упорядочивая, как он говорит, «хаос» Linux. Но что **Ян Мердок** думает о работе Debian – проекта, созданного им самим?

Ян Мердок [Ian Murdock] впервые затронул Linux, когда разработал манифест Debian и создал дистрибутив Linux, известный как Debian, в 1993. На этом он не остановился, и в 1999 году основал Progeny, ИТ-компанию на рынке Linux. Сейчас работает главным технологом в новообразованном Фонде Linux (Linux Foundation; появился в результате слияния OSDL и Free Standards Group) [уже нет: пока верстался номер, Мердок перешел на работу в Sun Microsystems, – прим. ред.]; его роль – ни больше ни меньше, как обеспечить надежность и стандарты в ОС Linux OS. **Ник Вейч** встретился с Яном в Нью-Йорке и обрушил на него вопросы читателей *Linux Format*, страдающих Debian-фобией.

Linux Format: Вы довольны тем, во что превратился Debian?

Ян Мердок: В общем-то, вполне доволен. Ясно, что Debian совершил немало хорошего и оказал мощное влияние. Он решительно превзошел все мои ожидания. Но есть аспекты, несколько обескураживающие меня.

По-моему, в некоторых отношениях Debian упустил большую возможность. Если рассмотреть, как используется Linux, от крупных предприятий до другого края спектра, то Debian – один из трех крупных дистрибутивов во всем мире: два других – Red Hat и SUSE. И есть несколько мелочей, которые его тормозят. А большой недостаток – то, что у него никогда не было предсказуемого графика релизов. Когда выйдет следующая версия? «По мере готовности» – это не самый притягательный ответ для широкого рынка, верно?

Полтора года назад я сказал, что перед Debian открываются широкие возможности, и что важнее всего для этого вовремя выпустить Etch. И вот мы видим не только провал, но и образ действий, который к нему привел – а именно, кое-какие разработчики предсказывали, что все это работать не будет, потому что деньги вложены в формулу проекта Dunc-Tank [группа оплачиваемых разработчиков дистрибутива Debian, – прим. пер.], обреченного на провал, и сами же пассивно-агрессивно вели себя так, чтобы это предсказание оправдалось, иллюстрируя худшие свойства процесса разработки программ с открытым кодом. Не говоря уже о том, что они [Debian] теряют отличные возможности.

»

LXF: Непохоже, что в других дистрибутивах столько внутренних раздоров.

ЯМ: В какой-то мере они довольно типичны. Такие раздоры можно наблюдать в любой фирме – вопрос в том, до какой степени они выплескиваются на публику. По большей части вы о них и не подозреваете, потому что все разборки происходят за закрытыми дверями. Так что на самом деле ничего нового тут нет. Просто в Debian все настолько открыто и прозрачно, что все на виду.

Я думаю, что в какой-то мере Debian овладел процессный амок. Должен существовать некий процесс, подлежащий масштабированию. Я хочу сказать, вы начинаете с одного человека, потом доходите до десяти – и перед вами проявляется новый набор проблем. Затем вы переходите от десяти к сотне, от сотни к тысяче, и следует выстраивать процесс в соответствии с его масштабом. Проблема слишком обширного процесса состоит в том, что разработку выполняет комитет – то есть нет сильного лидера, и никто не вправе взять принятие решений на себя. А иногда нужно принимать непопулярные решения; для этого и служат лидеры. В итоге такой комитет приходит к тому, что никто не отваживается принимать решения, пока с этими решениями не согласятся все. Но ведь организация разрастается, и всех удовлетворить получается невозможно, поэтому решения не принимаются вообще. Вот в таком тупике оказался сегодня Debian.

LXF: Если Debian выдохся, что тут исправишь?

ЯМ: Честно говоря, я такое уже вижу на Slashdot …так что я мог бы продолжать! Я считаю, фундаментальной ошибкой стало принятие этого демократического процесса, которое произошло уже после моего ухода и которому я противился.

LXF: Так вы были против этого?

ЯМ: Да, я был против. В основном причина та же: по-моему, в открытых проектах , как в любом бизнесе, чтобы сделать серьезную работу, нужно сильное руководство. И я думаю, именно это одна из причин успеха Ubuntu: у них есть сильный лидер, причем облеченный властью. Просвещенный руководитель прислушивается к мнению сообщества и принимает его к сведению, и понимает, что ему случается принимать и плохие решения, которые надо будет пересматривать. Полагаю, что ратовавшие за чистую демократию в Debian в какой-то мере рассматривали это как социальный эксперимент – что произойдет, если все решения ставить на голосование. Знаете, демократия в чистом виде… Она намного лучше выглядит на бумаге, чем получается на практике. Поэтому-то я всегда и был против. Знаете, меня порадовало нынешнее руководство Debian: я думаю, Энтони Таунс [Anthony Towns] хорошо поработал, и он не боится принимать непопулярные решения, примером чего может служить Dunc-Tank. С этой точки зрения, это в большей степени проблема организации. Будем надеяться, что сильное руководство будет продолжаться.

ОБ ОРГАНИЗАЦИИ DEBIAN

«Вы знаете, чистая демократия выглядит на бумаге куда лучше, чем на практике.»

LXF: Вы упомянули Ubuntu. Многие относятся к нему, как к «доведенному до ума» Debian. Вас не расстраивает, что Ubuntu – это фактически популистская версия Debian?

ЯМ: Позвольте мне теперь заступиться за Debian, после моих на него нападок. Нельзя забывать, насколько Debian был передовым и инновационным. Вся идея открытой разработки, распределенной разработки… это была модель, впервые примененная именно в Debian. Лавры первопроходца принадлежат Linux, но Debian стал первой попыткой создать что-либо таким способом.

Естественно, что разрушая старые основы и осваивая новое, совершаешь ошибки; успешность организации определяется, среди прочего, способностью осознать совершенную ошибку. Утверждение, что Ubuntu – это правильно сделанный Debian, отчасти подразумевает, что Debian сделан неправильно, а это уже, по моему мнению, заблуждение. А может ли Debian чему-то научиться у Ubuntu – его исключительной восприимчивости и всему хорошему, что в нем, если честно признать, есть? Разумеется. Я думаю, это единственное положительное влияние Ubuntu.

Ubuntu определенно поднял планку. Он оказал громадное влияние на множество людей во всем мире, которые используют Debian (а я считаю, что Ubuntu – это Debian). Полагаю, что ошибки они тоже делали, но они показали свою способность признавать просчеты и исправлять их. Еще раньше я достаточно громко выражал свое мнение по поводу совместимости, а именно: начали появляться пакеты Debian, которые в Debian невозможно запустить. Я помню, как это случилось с RPM в конце девяностых, и я не хотел, чтобы подобное произошло с Debian. Мы расходились во мнениях по поводу того, как делать разные вещи – DCC и прочее, возникавшее и исчезавшее. Но в конечном итоге, в настоящее время Ubuntu является активным участником моего текущего проекта, Linux Standard Base (LSB), ядром которого является совместимость, так что мои ранние опасения как раз и приняты во внимание.

LXF: Это приводит меня к следующему вопросу: не настанет ли момент, когда Linux Standard Base будет лоббировать какой-то конкретный менеджер пакетов?

ЯМ: Это имеет смысл, если люди просят. Но надо постоянно помнить, как работает LSB: мы действуем в мире, где уже существуют многочисленные реализации. Нельзя просто придти и заявить: «Тот прав, а этот ошибается». Скорее, вы усаживаетесь рядом с человеком и говорите: «Мы ведь все согласны с тем, что должны в чем-то придти к общему мнению, так почему бы вам не согласиться со мной, и все будет отлично?» А он в это время сидит рядом с вами и думает в точности то же самое…

Что же касается пакетов – да, некий стандарт пакетов определенно необходим. Взгляните на станицу закачки любого приложения Linux – ну, хоть MySQL: вы увидите там 15 разных пакетов. RHEL4-версия, RHEL5-версия, Debian-версия, каких только версий нет; и в какой-то степени LSB создана для того, чтобы решить проблему создания приложений таким образом, чтобы приложение работало в разных дистрибутивах; но она совершенно не касается того, как эта программа фактически появляется в системе. Поэтому мы недавно начали работу над стандартом пакетного менеджмента, и главным будет подход к этой работе. Мы пришли к дистрибутивам, потому что осознали: если мы хотим, чтобы стандарт имел вес во всем мире, дистрибутивы должны не заглатывать его без разбора, а открыто применять, в противном случае…

LXF: В этом нет смысла.

ЯМ: Да! Получится, что мы разработали стандарт, которым никто не пользуется, и, значит, зря потратили время. Поэтому мы собираем всех вместе и говорим: «Вот проблема, на которую надо обратить внимание», и по большей части все соглашаются. Обычно на этом месте все и распадается, потому что люди приходят и говорят: «Очевидно, чтобы продвинуться вперед, надо…» Обычно предлагаемое решение подразумевает необходимость все переписать.

LXF: Это решение а-ля Канберра, да? Невозможно выбрать между двумя одинаково хорошими вещами, и выбирают нечто слегка подмоченное.

ЯМ: Если бы! Выбрать – это еще полбеды, но ведь не выбирают, а создают что-то новое. Что касается стандартизации LSB, нам приходится работать в сильно рас-

пределенном, многополюсном мире, в котором способ создания стандарта так же важен, как его поддержание.

LXF: А нету ли опасности того, что приняв нечто как стандарт, мы утратим стимул к усовершенствованию?

ЯМ: Здесь мы приходим к вопросу взаимоотношений стандартов и инноваций. Обеспечивать стандарты вовсе не означает тормозить инновации, и на самом деле расширить инновацию можно, стандартизируя как можно меньше. Так что применительно к пакетам, не слишком важно, что все устанавливают пакеты в одинаковом формате, не слишком важно и то, чтобы при наличии двух пакетных систем обе обладали одинаковыми способностями и одинаковым интерфейсом. А важно вот что: чего потребителю технологии нужно от этой технологии? В случае с пакетами, об этом следует спрашивать у потребителей – разработчиков программ, которые желают создать единое приложение Linux, способное работать в любом дистрибутиве. Они говорят: «Слушайте, мы уже поддерживаем две, три, четыре разных платформы, и нам неохота поддерживать еще одну – нам не нужно RPM. Мы потому и пишем скрипты оболочки, чтобы просто загружать файлы в файловую систему. И нам годится расширяемое решение, которое позволит нам добавлять несколько маленьких строчек к коду, который у нас есть на сегодняшний день. Не вынуждайте нас вышвыривать все наши наработки в окно только потому, что вам не нравится предыдущая технологическая итерация». На самом деле все очень просто: стандартизировать как можно меньше и не вступать в дискуссию, имея некое предвзятое мнение о конечном результате.

LXF: Каковы, по-вашему, основные проблемы, стоящие перед LSB?

ЯМ: Основной проблемой, стоящей перед LSB, является сама по себе сложность работы, которую нам приходится выполнять. Если подумать о платформе Linux, она вовсе не так уж хорошо сгруппирована. Фактически, существуют десятки, или сотни, или тысячи различных компонентов, поддержкой которых занимаются самые разные люди, с самыми разными приоритетами. Кто-то понимает важность совместимости, кому-то до этого и дела нет – они хотят менять все, что им хочется, и когда им хочется.

Наша работа в том, чтобы упорядочить хаос – или, по крайней мере, скрыть его. Есть люди, которым подавай Linux как единую платформу, а тот факт, что в нем около 100 разных дистрибутивов, и что иногда какой-нибудь мелкий компонент разрушает всю совместимость в силу некой таинственной причины… это заставляет таких людей думать: «А вообще, жизнеспособен ли Linux?»

Проблема, стоящая перед нами – чтобы все эти люди с такими разными интересами стремились к одной цели.

LXF: Зачем?

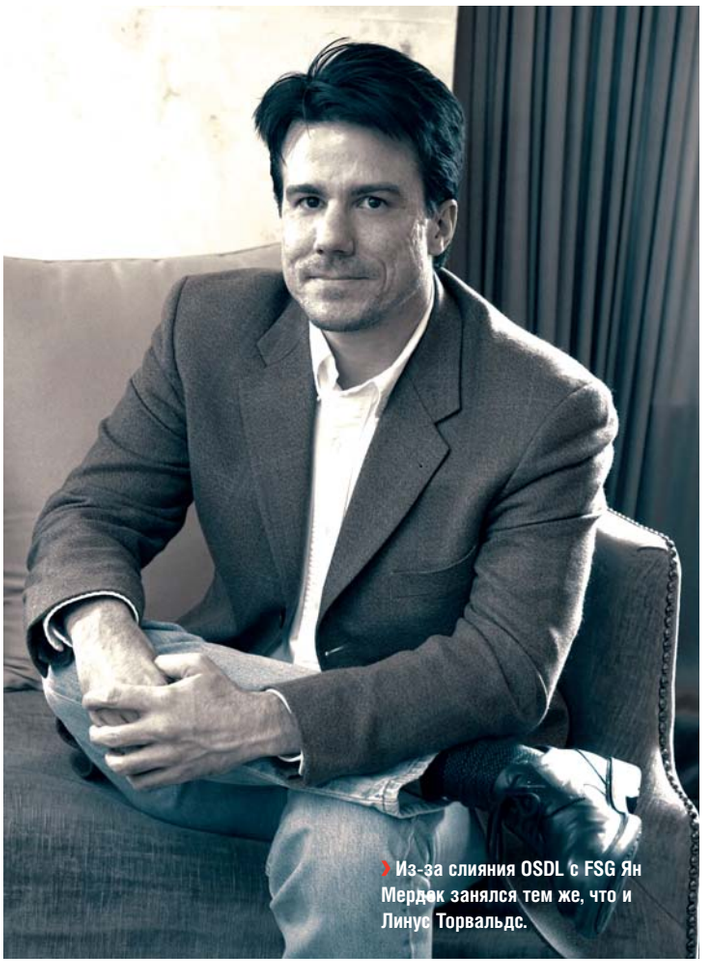
ЯМ: Потому что по мере роста Linux все больше людей сталкиваются со все более болезненными проблемами, которые мы пытаемся решать. Всегда трудно убедить людей начать лечить болезнь, которая еще не началась. «Но болезнь обязательно начнется через несколько лет, доверьтесь нам, мы собираемся предотвратить ее». Людей гораздо проще убедить заняться лечением, но не профилактикой.

LXF: Слияние OSDL с FSG облегчит ситуацию?

ЯМ: Да, определенно. Что касается FSG и OSDL, деятельность этих организаций в значительной степени пересекалась. В уставах этих организаций, на бумаге, никакого пересечения деятельности не было, но на самом деле оно было. Мы занимались LSB, а OSDL занимались Carrier Grade Linux [CGL] и Portland, которые весьма похожи на стандарты. В какой-то мере мы конкурировали друг с другом, а в этом совершенно не было смысла, потому что основа у нас у всех одна. И соединение этих организаций – да любая консолидация – послужит увеличению эффективности.

LXF: Вы считаете, что это также позволит прояснить цели?

ЯМ: Да, в том смысле, например, что вместо того, чтобы распыляться на LSB, CGL и Portland, мы сможем объединить их под одной идеей, а именно идеей стандартизации платформы Linux. Это повлечет за собой работу над компилятором, инструментарием, ядром, рабочим столом, некоторыми вертикальными функциями, например, телекоммуникационными, которые нацелены на определенную сферу.



Из-за слияния OSDL с FSG Ян Мердок занялся тем же, что и Линус Торвальдс.

Нам надо придумать четкое пояснение того, чем мы занимаемся. Люди обычно не задумываются об организациях, занимающихся стандартами. Они не думают «Какими стандартами занимается эта группа?» Они думают так: «А что это за штуковина такая – Linux, и как бы мне в него залезть?», «Как мне создать свой продукт для него?», «Как мне использовать его в моем предприятии?» – и далее в том же роде. Нам надо говорить на их языке – а помимо этого, просто засучить рукава и делать то, что должно быть сделано.

LXF: А вас не беспокоит, что в некоторых стандартах будут использованы патенты?

ЯМ: Об этом приходится беспокоиться всем, кто занимается стандартами.

LXF: Потому что так уже случалось, причем многократно.

ЯМ: Да, я полагаю, это уже проверенная временем технология: принять участие в разработке стандарта и внести свой вклад. Это называется подводными патентами: такие штуки скрываются где-то на заднем плане. Вы уже заняли нужную позицию, закрутили болты… Поскольку это – проверенная временем технология, организации, занимающиеся стандартами, разработали свою политику, а также и политику отношения к собственности, которая как раз и создана для подобных вопросов.

Мы – не исключение. У нас тоже есть политика IP, созданная для того, чтобы не возникали проблемы подобного рода. Конечно, наша политика отличается от практикуемой в типичных организациях, занимающихся стандартами, потому что мы – Open Source, но все же в большинстве случаев правила те же. LXF

О ПАКЕТАХ RPM И DEB

«Стандарты необходимы. Взгляните на страницу скачивания любого приложения Linux.»

Узнайте больше!

Ян обсуждает поддержку HP для Debian и проблемы, ожидающие Ubuntu, на www.linuxformat.co.uk/mag/murdock.html.

Что за штука...

libferris?

Согласно философии Unix, все в мире – файлы, верно? Но на самом ли деле все? Автор *libferris* **Бен Мартин** тоже задался этим вопросом.

Почему вы назвали библиотеку libferris?

И вы хотите знать, почему Бьюллер? [Феррис Бьюллер (Ferris Bueller) – название американской комедии, – прим. перев.] Нет, он тут совершенно ни при чем! Настоящая (ну, почти) причина выбора этого названия – слово "ferris" звучит похоже на акроним Virtual File System (VFS), если его произносить вслух: «Ви-Эф-Эс».

Э-э... а что такое виртуальная файловая система?

Идея состоит в том, чтобы работать с вещами, которые обычно не рассматриваются как файловая система, так, будто это она и есть. Старый добрый пример такого подхода – рассматривать FTP-сервер как локальный диск.

И что в этом хорошего?

Ну, например, представьте, что вы можете использовать некоторый запрос, и получить все файлы, удовлетворяющие этому запросу, сформировав содержимое виртуального каталога. Вы просто говорите «Найдите мне все файлы группы R.E.M.» и получаете каталог со всеми этими файлами. Поскольку это файловая система, то вы с этим каталогом можете делать то же, что и с обычными файлами – копировать их в плейер, ставить в очередь воспроизведения *Amarok* или что там еще можно делать с набором файлов.

Но есть уже GnomeVFS и KIO от KDE, неужели миру мало виртуальных файловых систем?

Главная идея libferris – файловой системой может быть что угодно, надо только предусмотреть в libferris подключаемый модуль для каждой новой штуки, которую вы хотите смонтировать. Такие модули уже существуют для реляционных баз данных, XML-файлов, почтовых ящиков, приложений и даже для расположения файловых систем поверх самих себя. Даже один файл, вроде RPM, может стать файловой системой. Извлечение и сохранение метаданных тоже входит в задачи проекта.

Погодите минутку: а как приложение может быть файловой системой?

Многие приложения ориентированы на данные. Например, в *Firefox* есть список окон и вкладок, и каждая содержит документ с иерархической структурой. Проигрыватель видео и музыки, например, *XMMS* или *Amarok*, содержит список файлов и текущую позицию воспроизведения в этом списке. Если вы смонтируете *XMMS* как файловую систему, то сможете копировать файлы из вашего списка воспроизведения на мобильный плейер с помощью обычной утилиты *cp*. Если вы смонтируете *Firefox*, то сможете скопировать на диск виртуальный файл, соответствующий третьему параграфу просматриваемой web-страницы.

Точно так же, как вы управляете ядром Linux через файловую систему **/proc**, вы сможете через некоторые файлы управлять поведением приложения. Например, записав единицу в файл **amarok://control/play**, вы даете команду начать проигрывание. А чтение и запись файла **amarok://control/volume** позволит вам контролировать громкость.

Чуть раньше вы упоминали «извлечение и сохранение метаданных». Зачем это нужно в libferris?

Когда вы упоминаете о метаданных для файловой системы, большинство думает про такие вещи, как размер файла, время модификации и имя владельца. Но в libferris метаданные могут включать любую информацию о файле. Например, для аудио-записей метаданными могут быть имя артиста и название композиции, извлеченные из ID3-тэгов. Поскольку все метаданные доступны через один и тот же интерфейс, вы может легко отсортировать каталог по ID3-номеру дорожки точно так же, как и по имени.

Понятно... А как насчет сохранения?

Метаданные могут храниться или с использованием механизма расширенных атрибутов на уровне ядра (Extended Attributes, EA), или в персональном хранилище в формате RDF. RDF – это набор стандартов W3C, позволяющий программистам указывать метаданные, которые могут быть использованы различными приложениями. Обычно так хранят ссылки на дополнительную информацию – например, на web-страницы или локальные файлы. Использование RDF для хранения метаданных позволяет добавлять заметки к FLAC-файлу, к NFS-тому, доступному только для чтения, к строке в базе данных или к web-странице. Использование метаданных возможно где угодно, так что вы всегда найдете файлы на их основе.

Идея в общем хорошая, но команда find на 500 ГБ диске работает довольно медленно.

В последние годы (а проект стартовал в 2001 г.) в libferris были добавлены возможности индексирования и поиска. Индексирование вынесено в отдельные модули, так что у вас есть возможность выбирать, с помощью чего создавать и хранить индекс. Есть модули для *PostgreSQL*, *Lucene*, *Xapian*, *Beagle* и других библиотек. Индексирование – достаточно заковыристая задача: и на скорость поиска по вашему индексу, и на объем используемой при этом памяти влияет способ обновления индекса. Наличие нескольких подключаемых модулей позволяет выбрать вариант, который вам подходит лучше всего.

Кстати, это значит, что вместо стандартной утилиты *find(1)* лучше использовать *ferris-find(1)*. У нее тот же самый интерфейс, но она гораздо быстрее, так как использует индекс файловой системы.

Ага, значит "*ferris-find(1)*"? Это значит, что файловая система *libferris* должна обслуживаться специальными утилитами?

Не обязательно. Вы можете подключать файловые системы libferris к ядру Linux с помощью Fuse.

Супер! Кстати, а что такое Fuse?

С помощью Fuse (Filesystem in UserSpace) вы можете заставить ядро Linux перенаправлять системные вызовы к приложению, запущенному пользователем, вместо того чтобы обрабатывать их на месте. Это позволяет разработчикам писать файловые системы, независимые от ядра, поскольку они работают не внутри него, а в виде отдельных процессов. Если я использую команду *cat*, чтобы посмотреть содержимое файла в файловой системе Fuse, *cat* попросит ядро Linux передать ему файловые данные, ядро же в свою очередь запросит процесс пользователя и вернет полученные от него результаты команде. Такая схема работы позволяет *cat* считать, что она работает с обычной ядерной файловой системой.

А при чем тут libferris?

У libferris есть команда *ferrisfs*, использующая Fuse для доступа через ядро Linux ко всему, что является файловой системой libferris. Это дает возможность программам, которые ничего не знают про

libferris, читать и писать данные в файловой системе libferris обычным способом.

Большая часть дистрибутивов Linux уже включает пакеты с Fuse и его утилиты.

Кроме того, существуют варианты реализации традиционных утилит командной строки, таких как *ls*, *cp*, *mv*, *rm* и *cat*, которые используют libferris без помощи Fuse, напрямую.

Зачем же возиться с переписыванием *ls*, *cp* и остальных, если libferris можно использовать через Fuse?

Ну, во-первых libferris была создана до того, как появился Fuse. А во-вторых, вы можете добавить парутройку очень удобных фильтров и опций сортировки при использовании утилит, разработанных с учетом libferris. Например, вы может использовать ввод-вывод с отображением файла в память, чтобы скопировать файл. Это может быть полезно, так как libferris сообщит ядру, что вы будете читать и писать последовательно, и ядро сможет подстроить стратегию кэширования. Таким образом, в вашем дисковом кэше останется гораздо меньше ненужной информации, чем при использовании других утилит копирования.

Работать с метаданными тоже гораздо удобнее при помощи новых клиентов. Вы можете указать, какие из них вы предпочитаете видеть при просмотре каталога в *ferrisls*, дополнительно указав отфильтровать или отсортировать результаты на основе метаданных.

Есть ли еще что-нибудь интересное для меня, о Создатель?

Если бы мне пришлось выбирать любимую возможность libferris, я бы указал на оповещения об изменении виртуальных файлов внутри XML-документа или базы данных *Berkeley DB4*. Одним из процессов вы можете смонтировать XML-файл и затем получать оповещения об изменениях, выполняемых другим процессом. Это делает поведение DB4-файла очень похожим на поведение настоящей файловой системы.

Второе место получило бы сохранение истории обращения к метаданным. Потом вы можете выполнять к этой истории запросы типа «Какие последние 50 фильмов я смотрел?» или «Какие DVD я смотрел на неделе после возвращения с Linux Kongress?»...

Какие еще безумные планы есть для libferris?

Подключение новых штук у нас всегда в поле зрения. Мы также планируем интегрироваться с большим числом XML-утилит и стандартов: например, пользователи смогут выполнять запросы XQueries на файловой системе libferris или на ее индексе.

Ну вы и голова. Где я могу получить подробную информацию и скачать программу?

Спасибо, я знаю. Домашняя страница нашего проекта: http://witme.sf.net/libferris.web. Новые версии обычно выходят каждые один-два месяца, и анонсируются на http://freshmeat.net. LXF

Ультрасовременный Linux-компьютер

УЛЬТРА-КОМПЬЮТЕР под Linux

Четыре ядра, 4 ГБ памяти и 1500 ГБ места на диске. Но работает ли здесь Linux? Безусловно, работает... Не без **Пола Хадсона**.

Нас тошнит, когда говорят, что Linux не поддерживает современное оборудование. Нас тошнит, когда заявляют, что Vista – лучший выбор для продвинутых пользователей. Но больше всего нас тошнит от помешанных на скорости типов, покупающих самое лучшее «железо», чтобы гонять на нем Windows. А знаете, почему? Единственная настольная ОС, способная извлечь все преимущества новейшего и мощнейшего оборудования – это Linux.

Наше утверждение не голословно: мы решились доказать его, потратив свои кровные денежки. С кредитной картой компании наготове мы пробежались по магазинам и купили самый лучший компьютер, какой только можно было найти. Затем, что немаловажно, мы поставили на него Linux. Так вот, он работает! Итак, если вы затеяли собирать самый современный компьютер, совместимый с Linux, или решили обновить пару компонентов, и вам нужно взглянуть, как они приживутся, а то и просто хотите доказать своим друзьям, что Linux может надрать задницу самому крутому «железу», которое можно купить за деньги – читайте дальше.

Предпоследний Linux-компьютер

Когда компьютерщики хотят повеселиться, они вспоминают знаменитые слова Билла Гейтса, что «640 КБ хватит любому». Кстати, на самом деле он так не говорил, это лишь расхожая легенда; но люди, покупая компьютеры, допускают подобную ошибку постоянно. Фактически, это случилось и с нами, когда мы собирали Linux-суперкомпьютер в прошлый раз, в 2004 г. (см. LXF53). Тогда 64-битная технология только-только дошла до настольных систем, и мы думали, что 1 ГБ ОЗУ – предел мечтаний на все времена. Но с тех пор виртуализация далеко опередила наши предсказания: теперь можно запускать Linux под Linux или Windows под Linux без дополнительных вложений, и для серьезных пользователей компьютера 1 ГБ – абсолютный минимум.

Другой большой скачок вперед – переход на многоядерные процессоры. В 2004-м ядро Intel Prescott было невероятно мощным процессором, способным взять барьер в 3 ГГц, если вас не пугал теплоотвод размером с ладонь. В то время на роль новейшего компьютера годился AMD: мы выбрали Athlon 64 3400+, 64-разрядную мощь, упакованную в холодный и тихий чип. Но сейчас, благодаря своей архитектуре Core, Intel снова вырвался вперед. Если вдаваться в подробности, то это сильно переработанный (в плане энергосбережения) Pentium III под маркой Centrino; в итоге Intel смог без особых усилий уместить два таких ядра в одном процессоре. Недавно снова произошло удвоение, и теперь Intel предлагает 4-ядерные процессоры, умещающие более чем 10 ГГц вычислительной мощи на одной подложке.

В итоге этого прогресса наш былой суперкомпьютер на Linux превратился в довольно отсталый. Если вы хотите запускать игры типа Quake 4 с высоким разрешением, видеокарта GeForce 5950 Ultra – а четыре года назад, когда ее выпустили, это был последний писк – уже не для вас. Так что же тогда вам нужно? »

Список покупок-2007

Мы собираемся найти лучшие комплектующие, которые только можно купить, и скрепить все это превосходным Linux-клеем. Приступим...

Мани-мани-мани...

1 фунт стерлингов – это чуть меньше 2 долларов США; здесь и далее приводятся цены комплектующих на рынке Великобритании. Стоимость их в России зависит от места вашего фактического проживания.

Где мы покупали

Все наши цены получены с www.scan.co.uk, который предоставил все компоненты для этой статьи через день после того, как мы сделали заказ. Все цены в тексте включают 17,5% НДС, хотя, чтобы помочь вам, на стр. XX мы даем расценки без налога.

Процессор

Вычислительные мощности еще никогда не стоили так дешево: двуядерные чипы 1,8 ГГц с 2 МБ интегрированного кэша и внешней шиной с частотой 800 МГц уже сейчас можно найти за £100 (Athlon 64 3400+, с тактовой частотой 2,2 ГГц, в нашем предыдущем суперкомпьютере стоил почти 300 фунтов). Даже если вам позарез нужна шина 1066 МГц и более высокая тактовая частота, 200 фунтов обеспечат вам 2,4 ГГц двухядерный чип с 4 МБ интегрированного кэша – считай, 100 фунтов остались в кармане!

Код Linux очень хорошо работает на нескольких ядрах. Философия Unix «множество инструментов, хорошо делающих одно дело» означает, что задания очень легко распараллеливать; возможное исключение – задачи, интенсивно нагружающие процессор, типа кодирования или сжатия видео. Как AMD, так и Intel (а также ряд сторонних разработчиков ПО) ведут интенсивную работу по гладкому распределению программ между ядрами, так что мы убеждены, что двухъядерные чипы – это просто переходный этап между одноядерными и многоядерными.

Отчасти потому, что, по нашему мнению, двуядерные чипы явились в мир ненадолго, а отчасти потому, что процессоры сейчас подешевели, мы гульнули на всю катушку и вбухали львиную долю наличных в лучший процессор для настольных систем, который можно было купить: Intel QX6700 – 4-ядерный Core 2 Extreme, работающий на частоте 2,66 ГГц, с 1066 МГц шиной и 8 МБ интегрированным кэшем. На момент подготовки статьи этот малыш отобрал бы у вас нехилые £600, но цена, вероятно, несколько упадет к тому времени, как журнал попадет вам в руки.

Материнская плата

Грех выбирать самый лучший процессор, не подкрепив его аналогичной по качеству материнской платой, и сейчас это означает поиск платы, построенной на превосходном чипсете от Nvidia – Nforce 680. Поддержка 680-го в Linux не слишком хороша, особенно в старых версиях ядра, но мы готовы были рискнуть некоторой нестабильностью работы, учитывая, что множество людей работают над улучшением ее поддержки в новых дистрибутивах Linux. Наш процессор поддерживает внешнюю шину 1066 МГц, так что нам требуется и совместимая «мамка» – а это дополнительные расходы. Nforce 680 использует сдвоенные коннекторы Ethernet для обеспечения максимальной пропускной способности и надежности. Если вы намерены связаться с 680-м, наш совет – соблюдайте осторожность: припасите лишнюю Ethernet-карту с разъемом PCI!

Нашим выбором стала прекрасная карта IN9 32X-MAX от Abit, поддерживающая SLI (комбинация двух видеокарт для увеличения графической мощи – Linux поддерживает SLI неустойчиво, но со временем это, конечно, будет исправлено) и встроенный Wi-Fi. Она с запасом проработана по инженерной части и предоставляет максимальную свободу действий для разгона, если вы пойдете этим путем. В этой плате нам особенно понравилась реализация сброса CMOS BIOS'а: если вы попытаетесь разогнать свою машину (а почему бы и нет – это же так весело!) и загоните ее слишком далеко, вам не придется возиться с перемычками, возвращая установки по умолчанию. Вместо этого просто щелкните переключателем на задней панели компьютера – и вернетесь к безопасным установкам.

Оперативная память

Когда мы последний раз собирали Linux-компьютер высшего класса, мы выбрали оперативную память Corsair Memory, из-за репутации этой компании как производителя отличных и самых быстрых чипов ОЗУ. На сей раз мы сделали тот же выбор: новая серия Corsair Dominator объединяет низкую латентность с эффективным теплоотводом и высокой пропускной способностью. Все зависит от необходимой скорости работы памяти: приходится идти на компромисс между быстродействием ОЗУ и временем запаздывания, а все вместе стоит по-настоящему больших денег. Основываясь на своем опыте и утечке информации от редактора лаборатории PC Format, мы урвали 800-МГц память со значением CAS Latency, равным четырем. Два модуля по 1 ГБ стоили 170 фунтов, то есть 4 ГБ памяти можно прикупить за 340 фунтов. Если у вас есть лишние деньги, и вы хотите достичь наинизшего возможного запаздывания, два 1 ГБ модуля Dominator с CAS Latency, равной 3, потянут в два раза дороже. 1066 МГц модули с таким же запаздыванием обойдутся вам в 440 фунтов, и мы не очень-то уверены, что прирост скорости будет заметным. Тем не менее, и плата, и процессор уже готовы работать на частоте 1066 МГц, если латентность и цены понизятся, так что неплохо иметь запас прочности.

Ультрасовременный Linux-компьютер

Видеокарта

Какое-то время Nvidia и ATI шли по производительности голова к голове, но в последнее время Nvidia вырвалась в лидеры, со своей новой 8000-й серией процессоров GeForce. Nvidia всегда славилась прекрасными драйверами для Linux, что сразу дает ей преимущество перед ATI, где поддержка Linux отнюдь не блещет.

Прежде чем выкладывать свои кровные за видеокарту, обязательно примите в расчет размер монитора. Мы выбрали Dell 2407WP, 24-дюймовый экран с родным разрешением 1920x1200. Хотя 8800 GTS и годится для игр при таком разрешении, его старший брат GTX выдает на 20% больше кадров в секунду, обеспечивая дополнительные возможности по настройке полноэкранного сглаживания. Наш монитор имел время отклика 6 мс и контраст 1000:1, так что мы взяли видеокарту, позволяющую насладиться всеми его преимуществами: XTX 8800GTX с 768 МБ памяти на борту.

Блок питания

Оборудование нашего компьютера потребляет изрядную мощность. Можно просуммировать запросы всех компонентов и взять БП ровно с таким числом ватт, но лучше иметь некоторый запас для расширения на будущее. Как ни странно, между разными блоками питания есть множество отличий: у одних большие вентиляторы – они могут вращаться медленнее, чем маленькие, и потому меньше шумят; у других больше коннекторов для внутренних устройств; а лучшие из них надежно защищают от скачков напряжения и гарантируют устойчивый ток. Мы выбрали БП XClio 1000W StablePower, поскольку он способен поддерживать GeForce 8800 и 4-ядерный процессор. Цена ему 160 фунтов стерлингов – довольно дорого, но при подборе блока питания мы не рекомендуем гнаться за дешевизной.

Звуковая карта

Наша материнская плата Abit поставляется со встроенным звуковым контроллером, но взглянем правде в глаза: такие вещи хороши, если вы хотите лишь слушать музыку да играть в отстойные 2D-игры. Для полу-приличного 3D-звука и аппаратных эффектов единственным выбором уже давно является SoundBlaster. Более того, новые карты X-Fi Extreme этой компании стоят всего 30 фунтов и гарантируют радость для вашего слуха на многие годы. Да, мы знаем, что Linux-драйверов для X-Fi не будет еще по крайней мере месяц, но хотеть не вредно!

Устройство хранения

С тех пор, как мы собирали последний компьютер мечты с Linux на борту, диски не слишком изменились. «Родная» очередь команд идет сейчас как стандарт, и диски SATA, наконец, обзавелись функцией, долгое время бывшей преимуществом SCSI: способностью осмысленно переупорядочивать операции чтения для минимизации перемещений головок диска, что в целом увеличивает скорость работы. Остается лишь выбрать конкретную емкость – за 90 фунтов вы получите 500 ГБ дискового пространства, а если вы склонны к роскоши, то за 170 фунтов можно приобрести 750 ГБ диск.

Такие мы и присмотрели. Грубо говоря, 500 ГБ модели лучше по соотношению «цена/объем», и тех, кто не ведет фотолетопись и не редактирует домашнее видео, почти наверняка устроит один диск на 500 ГБ. Но мы возжелали заплыть за 1000 ГБ буек – наш новый компьютер укомплектован 1,5 терабайтами дискового пространства.

Корпус

Когда в прошлый раз мы собирали компьютер с нуля, нашим корпусом был Coolermaster. В этом году Coolermaster таковым и остался: качество его сборки неизменно впечатляет, подтверждая, что ко всем его зонам приложено немало усилий. Корпус для нашего крутого компьютера – без винтов внутри, и установить в нем карты и диски – раз плюнуть. У него есть также два USB-порта и один порт FireWire сбоку, и, самое главное, он потрясающе выглядит.

Клавиатура

К огорчению ярых MS-ненавистников, здесь, в Башнях LXF, мы расцениваем оборудование Microsoft как одно из лучших доступных пользователю. Мы решили остановиться на проводном устройстве – беспроводные прекрасны, если надо изредка понабирать текст, но нам-то приходится жать на клавиши очень часто. Вдобавок мы много играем в UT [Unreal Tournament], и нет ничего хуже, чем взорваться на части лишь потому, что ваша клавиатура залипла, как раз когда вы жмете Jump. Microsoft Natural Value Pack за 19,99 фунтов стерлингов завершает наш список компонентов; теперь мы готовы собрать все это вместе...

Скорая помощь

Материнская плата Abit IN9, которую мы выбрали, имеет огромное число опций BIOS, многие из которых связаны с разгоном. Позаботьтесь об установке лучшего из доступных вам кулеров, затем поиграйте с опциями uGuru в BIOS – и увидите, что это «железо» способно на гораздо большее, чем при своих установках по умолчанию!

Скорая помощь

Убедитесь, что выбранный вами БП имеет коннекторы SATA и PCI Express – оба наших жестких диска и DVD-привод имеют SATA-разъем, а карта GeForce нуждается в двух разъемах питания PCI Express, чтобы работать в 3D-режиме.

»

Пошаговые инструкции

Набрав оборудования, приступим к сборке нашего конструктора. Сейчас расскажем, как...

Сборка компьютера подобна игре в Кубик Рубика: пока вы не собрали его в первый раз, это кажется невозможным, но когда вам покажут решение, вы уже не понимаете, о чем тут было беспокоиться. Вот хорошая новость для вас: пока вы будете соблюдать осторожность и носить антистатический браслет, ваше оборудование без последствий стерпит множество надругательств, и даже если это ваша первая самостоятельная сборка ПК, бояться нечего.

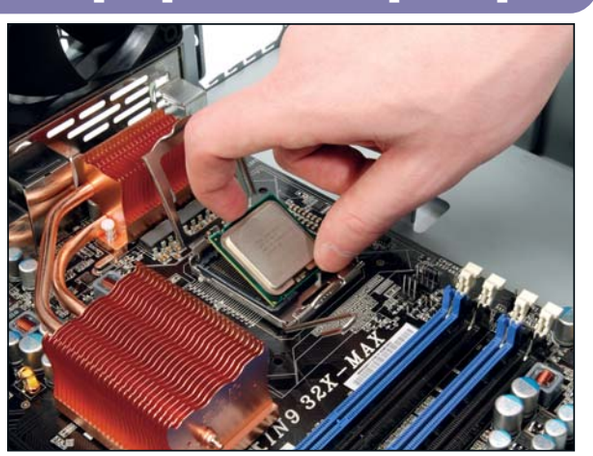
Если вы выбрали не совсем такие компоненты, как у нас, учтите это, прежде чем приступать к сборке. Например, наш 4-ядерный процессор использует Intel Socket T, выводы которого расположены на материнской плате, а не на самом чипе. Это снижает шанс случайно погнуть вывод, потому что на материнской плате они подпружинены. Если вы используете более старый чип или модель AMD, выводы будут на чипе, так что поаккуратнее там на шаге 2! Если у вас такой же процессор, как и у нас, обязательно сохраните одноразовую пластиковую упаковку с процессорного гнезда материнской платы: она понадобится, если нужно будет отослать материнскую плату назад для замены.

Убедитесь, что руководство к материнской плате будет у вас под руками на шаге 9, поскольку практически невозможно упомнить, куда идут все эти кабели от корпуса!

Шаг за шагом: Собираем суперкомпьютер с нуля

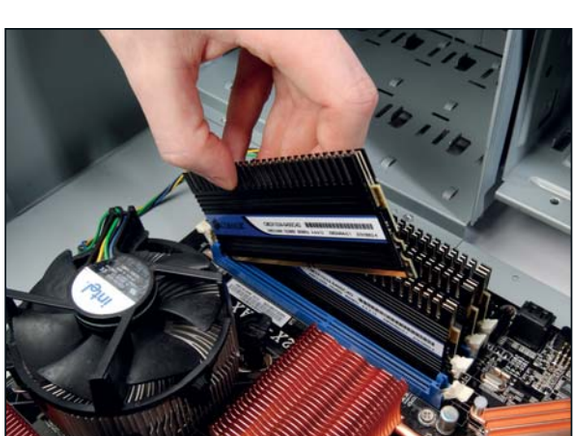
1 Установите материнскую плату

Вставьте металлические разъемы в заднюю панель вашего корпуса, затем пристройте материнскую плату и привинтите ее.



2 Вставьте процессор

Удалите защитную оболочку с вашего процессора и вставьте его. Нанесите немного термопасты на поверхность и установите вентилятор процессора.



3 Установите модули памяти

Четыре гигабайта памяти могут показаться избыточными, но в условиях изобилия виртуальных машин – чем ее больше, тем веселей.

4 Установите блок питания

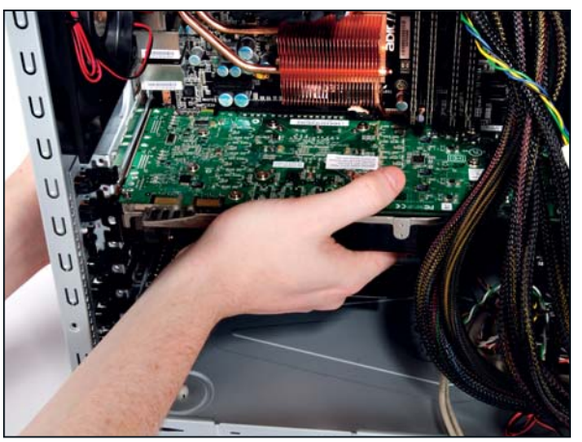
Установите свой БП так, чтобы вентилятор смотрел на материнскую плату. Убедитесь, что кабели уложены аккуратно.

5 Смонтируйте жесткие диски

Наш корпус оснащен безвинтовыми креплениями для жестких дисков, что упрощает их монтаж – они просто вдвигаются.

6 Вставьте звуковую карту

Установите вашу звуковую карту в самом низу материнской платы, чтобы оставить максимум пространства между ней и видеокартой.

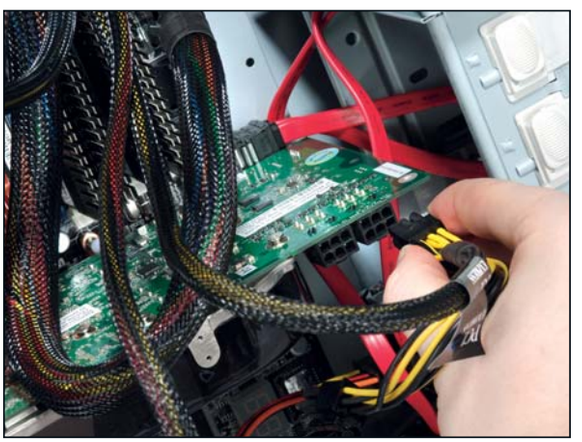


7 Установите видеокарту

8800 GTX занимает два слота и весьма тяжела, поэтому вставьте ее поглубже и затем убедитесь, что она прочно закреплена.

8 Вдвиньте оптический привод

Вдвиньте свой DVD-привод. Если у вас дешевый блок питания, его кабель может оказаться коротковат, и придется опустить устройство пониже.



9 Подключите кабели

Скучная часть – подсоединение всех кабелей. При каждой сборке ПК нам кажется, что кабелей стало больше, чем когда-либо, так что на это потребуется время.

10 Удивляйте своих приятелей

Готово! Можете обзванивать своих друзей и приглашать их подивиться.

Скорая помощь

Многие части вашего компьютера можно соединить только одним способом, но есть одно простое исключение – винты, которыми прикручивается материнская плата к задней стенке. Большинство материнских плат поставляются с кучей винтов, имеющих незначительные отличия в размере. Если вы воспользуетесь не теми, может оказаться, что материнская плата прикручена не совсем правильно, так что внимательно подбирайте винты!

Суперкомпьютер на марше

Тестируем совместимость с Linux, позволяя себе немного хакерства.

Ежедневная работа с последним ПО для Linux может с ума свести. Часами копаться в вещах, не входящих в ваши обязанности, каждому приятно, но не у каждого хватает времени и терпения, чтобы заставить все работать. Мы вооружились другой тактикой для этого компьютера: если что-то не начинало работать в течение 20 минут, мы переходили к следующему компоненту.

Пока мы это пишем, на очередные релизы Ubuntu и Fedora накладываются заключительные штрихи, так что мы ожидаем увидеть значительное число поддерживаемого оборудования «в следующей версии» – страшная фраза, которой боятся все пользователи Linux. Тем не менее, мы уверенно взялись за работу, взяв SUSE 10.2 DVD и большую банку кофе. Как мы и ожидали, SUSE прекрасно распознала все наше оборудование и менее чем за десять минут предоставила нам работающую настольную систему.

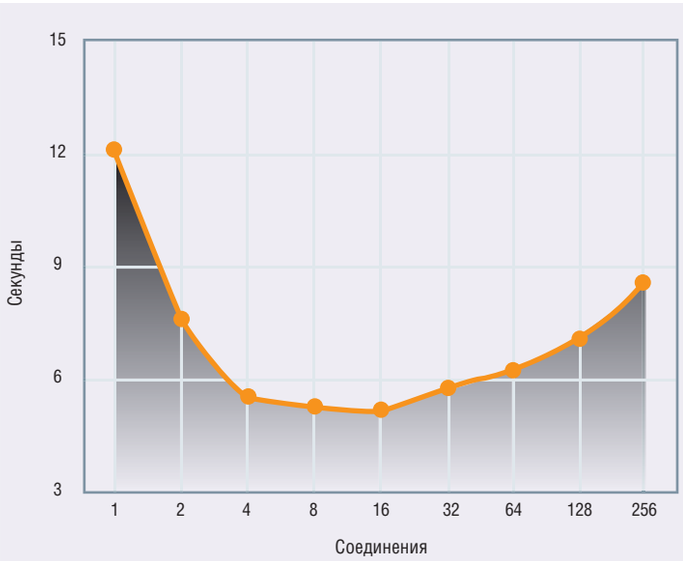
Но первой проблемы пришлось ждать недолго: хотя обе сетевые карты определились, они отказались отвечать на запросы DHCP. В Linux есть превосходный Ethernet-драйвер (*forcedeth*), официально поддерживаемый Nvidia, но он все еще находится в активной разработке. Решением нашей проблемы стала бы загрузка последнего ядра и его ручная компиляция, но это выводило нас за пределы двадцатиминутного лимита на решение проблемы, так что мы отложили ее на потом и в качестве временной заплатки подключили старую сетевую карту Intel; она сразу же заработала безупречно.

Поддержка звуковой карты

Поддержка двух звуковых карт поровну разделилась на «работает с пол-пинка» и «вообще не работает», что, похоже, в наши дни в порядке вещей для драйверов в Linux. На момент написания этой статьи, Creative не поддерживал X-Fi в Linux, а ALSA воротила нос от этой карты, фокусируясь вместо нее на устройстве Realtek HD, встроенном в материнскую плату. Оно заработало «из коробки» и – чтобы быть справедливым – превосходно звучало. Creative обещает полную поддержку X-Fi в Linux, включая аппаратную обработку звука и продвинутую

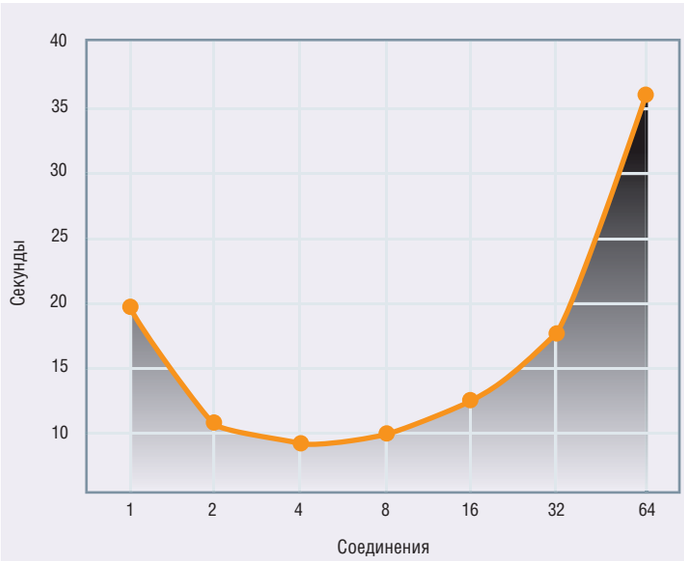
Графики производительности Linux-суперкомпьютера

Apache 2.2



› В этом тесте *Apache* обслуживал до 100 тыс. запросов HTML-файла размером 12 КБ с отключенной функцией *KeepAlive*. Небольшим узким местом здесь была ширина канала, то есть оптимальную скорость мы получили при 16 одновременных соединениях.

MySQL 4.1



› Здесь мы использовали *MySQL* для вставки 100 тыс. строк, их индексации и последующего выполнения 100 тыс. операций поиска. *MySQL* «проседала» при большом числе соединений, так что рекомендуем ограничить их количество примерно 50-ю.

систему объемного звука EAX, во втором квартале этого года. Только после этого мы будем по-настоящему счастливы...

Кроме этих двух проблем – а ведь мы предупреждали вас насчет Ethernet! – Linux прижился на этой машине достаточно хорошо. Теперь посмотрим, на что он способен...

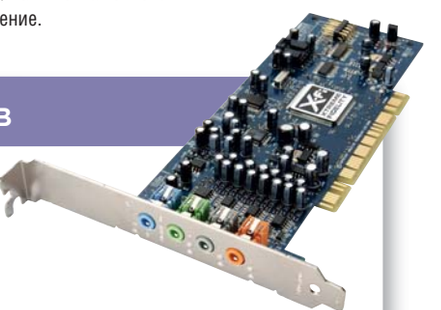
Серьезные тесты

В начале статьи мы объявили, что Linux – единственная настольная система, способная полностью использовать преимущества нашей машины, понимая под этим, что для него существует множество приложений, способных на параллелизм. Как Apache, так и *MySQL* включены как стандарт в большинство дистрибутивов, и хорошо распараллеливаются по мере добавления большего числа ядер. Но, как и в любом ПО, здесь имеет место падение ожидаемой эффективности из-за узких мест, проявляющихся по соседству. Идеальный вариант, как и ожидалось, получается при одном процессе на ядро – будь то соединение Apache, запрос к *MySQL* или поток кодирования OGG. Но, как показывают наши графики, чем больше потоков наваливается на машину, тем больше они конкурируют за ресурсы, что приводит к значительному замедлению.

Как можно видеть из графиков, многие из задач, которые могут быть распараллелены, выполняются на четырех ядрах в 2,2 – 2,7 раз быстрее, чем на одном, и это превосходный результат. На так называемых идеально распараллеливаемых операциях, типа использования кодера *Ogg Vorbis* для кодирования восьми WAV-файлов в OGG-формат, четыре ядра работали в 3,85 раз быстрее, чем одно. Это потому, что каждый поток кодирования может работать независимо от других, и данных для обработки настолько мало, что эта операция ограничивается только возможностями процессора.

Если вы давно читаете *Linux Format*, вам может быть интересно узнать, что мы откопали копию нашей старой программы тестирования, *LXF Bench 2004*, и смеха ради ее запустили. Результат? Компьютер-2007 выполнил его в 15,2 раза быстрее, чем наш эталонный ПК (Pentium 4, 1,8 ГГц), в многопроцессных тестах и в 5 раз быстрее в тестах ОЗУ. Для сравнения, «предпоследний» Linux-компьютер из LXF53 был соответственно в 3 и 2,75 раза быстрее, чем эталонный, так что наша новая сборка продемонстрировала существенное улучшение.

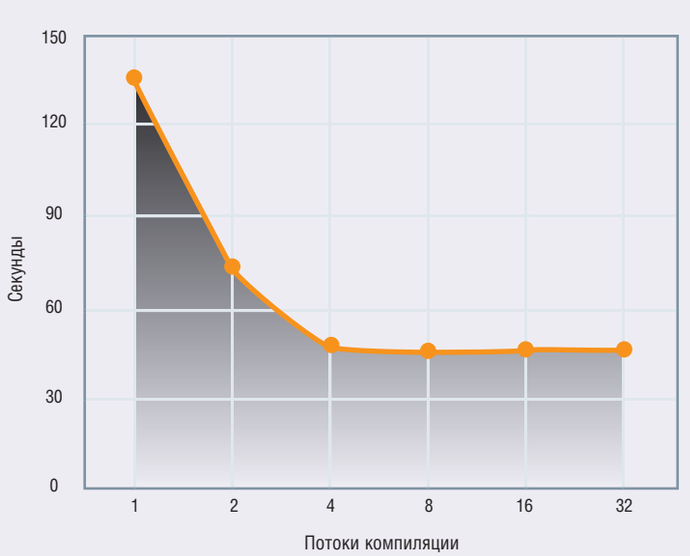
Поддержка драйверов



Хотя наша машина работала великолепно, нас расстроило отсутствие поддержки драйверов для периферии. Особенно досадно было то, что SUSE не сумела привести в чувство сеть – хорошо еще, у нас была запасная карта для решения этой проблемы. Да, мы знаем, что все будет исправлено в новом выпуске ядра, но это нас не спасет, поскольку нет простого способа внести исправления в имеющееся ядро.

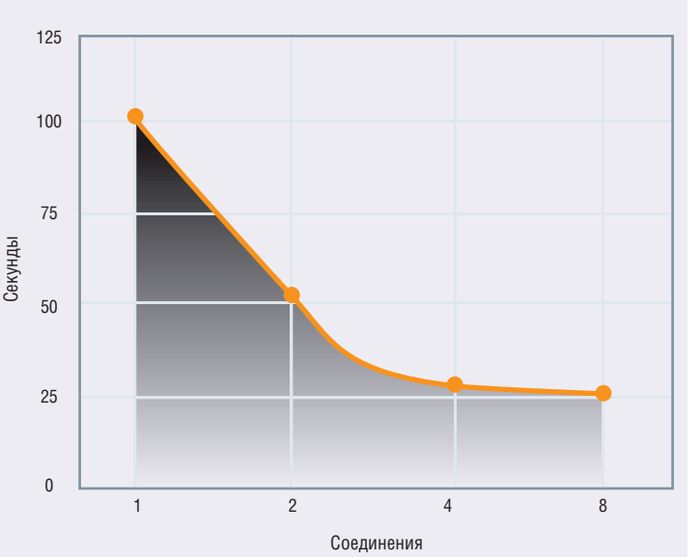
Помимо этого, разочарование было только одно: отсутствие поддержки для Creative X-Fi. Оно, правда, смягчилось тем фактом, что поддержка встроенного звука превосходна, и, конечно, Creative обещает сделать Linux-драйверы со дня на день. Но это не уняло нашу тоску по отсутствующим функциям, тем более что карта X-Fi выпущена сравнительно давно. Давай-ка, Creative, разберись с этим!

GCC



› Это сборка исходного кода PHP с помощью *GCC 4.1.2*. Компиляция зачастую хорошо распараллеливается и интенсивно использует процессор – не ожидайте значительного ускорения от добавления большего числа потоков компилятора, чем число имеющихся ядер.

Ogg Vorbis



› В этом тесте использовался *OggEnc* для кодирования восьми WAV-файлов по 45 МБ в формат OGG с уровнем качества 3. Узким местом здесь являются процессоры, и использование отдельных файлов гарантирует, что каждый процессор может работать на полную мощность.

Следующий рубеж

Аппаратные разработки для пользователей Linux, которым не терпится…

Оборудование всегда идет впереди, и инженеры, занимающиеся ПО, всегда рады отхватить больше процессорного времени, больше памяти и больше места на диске, чем когда-либо раньше. Хотя наша новая ультрасовременная машина не бессмертна, она проектируется с запасом, чтобы работать как можно дольше. Мы ожидаем, что на ней можно будет играть в последние игры с максимальным разрешением по крайней мере года три, и она должна удерживаться выше среднего в течение пяти лет.

Если вы ищете домашний компьютер, который выдюжит все, чем вы его нагрузите, плюс еще кое-что, то лучшего и не найти. Он не дешевый, но исключительно мощный, и верно прослужит вам несколько лет. Если вы ищете компьютер для Linux-сервера, построенный из лучших компонентов и способный обслужить до 100000 web-страниц за 5 секунд (понятно, если ширина канала позволит!), или сервер баз данных с огромным объемом кэшируемых и индексируемых данных, то это будет дешевым выбором – плюс он надежен и предоставляет запас прочности для расширения в будущем.

Видеокарта в этой машине достаточно хороша, чтобы еще долго радовать пользователей Linux, поэтому наш основной интерес насчет будущего связан с процессором и оперативной памятью. Intel медленно движется по направлению к системе межпроцессорного взаимодействия HyperTransport, которая обеспечит увеличение скорости многоядерных систем вроде нашей. Есть также надежда, что в их процессоры скоро будет добавлен интегрированный контроллер памяти, что устранит очередное «узкое место», возникающее, когда несколько ядер одновременно обращаются к ней.

Технология HyperThreading

Одним из вероятных побочных эффектов в будущих ядрах является возможность возврата к технологии HyperThreading – это решение симметричной многопоточности от Intel, позволяющее одному ядру исполнять два процесса одновременно. Каждое ядро с поддержкой HyperThreading отображается как два ядра: одно реальное (и способное работать со 100%-ной производительностью) и одно виртуальное (его производительность намного ниже). Именно здесь и возникает проблема – нам знакомы примеры, когда многопоточное приложение работает на одном реальном ядре и одном виртуальном, оставляя другие физические ядра в бездействии.

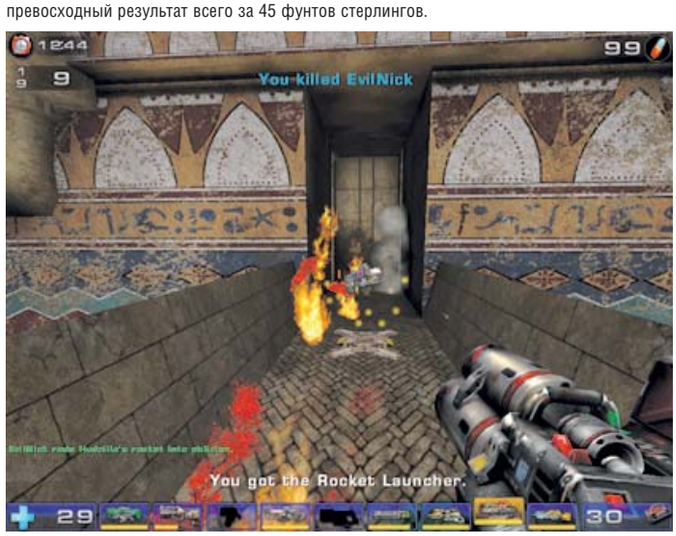
Что касается оперативной памяти, то виртуализация наконец-то широко распространилась: запустить альтернативные ОС на вашем компьютере ничего не стоит, и при объеме ОЗУ большем, чем 1 ГБ, каждый может насладиться свободой, которую предоставляет виртуализация. Мы ожидаем, что со временем виртуальные машины будут более тесно интегрированы, приближаясь к 100%-ной поддержке гостевых систем оборудованием, так чтобы поддерживались абсолютно любые устройства (включая 3D-видеокарты). Коль скоро вы можете выде-

Доблесть — играючи!

Пока разработчики игр не научатся, наконец, эксплуатировать многоядерные чипы, ни одна игра не сможет в полной мере насладиться преимуществами нашего Linux-суперкомпьютера. Тем не менее, важно знать, насколько вы сможете нагрузить свою видеокарту, так что мы запустили наш любимый тест: *Unreal Tournament 2004*. Со всеми настройками, выставленными по максимуму, и с разрешением 1920x1200, мы присоединились к интернет-игре с 16 игроками, чтобы протестировать машину. После нескольких отчаянных пробежек, прыжков, перестрелок и смертей, мы измерили, что *UT* работает в среднем на скорости 85 кадров в секунду – быстрее, чем могут воспринять глаза человека.

При переходе в *Doom 3* максимальное разрешение составило 1600x1200, что мы компенсировали увеличением сглаживания до 16x и установкой графики в высший уровень детализации. Даже это не смутило машину: демо-версия все время работала в среднем на 73 кадрах в секунду. Это куда больше, чем 60 кадров в секунду, необходимые для уверенной вертикальной синхронизации, устраняющей обрезание видео, которое в противном случае заметно на 24-дюймовом мониторе.

Запуск требовательных к мощности игр на длительное время, особенно когда вы размещаете на сервере LAN-игру, обнаружило одну небольшую проблему с нашим новейшим компьютером: когда процессор начинает нагреваться, его вентилятор становится слишком шумным. Если вы находитесь дома и компьютер стоит под столом, это может не сильно мешать вам. Но если вы находитесь в офисе или просто чувствительны к звуку, вы захотите выкинуть в канаву этот входящий в комплект вентилятор и найти что-нибудь получше. Конечно, положительной стороной здесь является то, что процессор остается холодным, что увеличивает его срок службы, так что к этому вопросу нужно подходить с умом. Мы поговорили с нашими друзьями из QuietPC (**www.quietpc.co.uk**), и они порекомендовали кулер Zalman с заковыристым именем CNPS9700. Через день он был у нас на столе: эдакий монстр, укомплектованный 110-мм вентилятором и десятками лопастей для увеличения площади поверхности. Но поставив его в машину, вы сразу же почувствуете глубочайшую разницу – даже при работе со 100% загрузкой машина остается тихой. Мы смогли отсоединить основной вентилятор корпуса, чтобы еще больше снизить уровень шума, и это превосходный результат всего за 45 фунтов стерлингов.



› **Даже при максимальной детализации *UT* работал просто сказочно.**

Ультрасовременный Linux-компьютер

лить ядро процессора для каждой запускаемой ОС, основным сдерживающим фактором для подвигов виртуализации становится память. Советуем запасти ее столько, сколько вы сможете себе позволить. Мы предпочли бы пожертвовать 300 МГц процессора и потратить сэкономленные деньги на дополнительную пару гигабайт ОЗУ.

Переход на многоядерные системы пойдет лишь на пользу Linux. Некоторые поставщики ПО пытаются скорректировать свои модели лицензирования, взимая со своих потребителей плату с ядра, а не с процессора, а Linux свободен в использовании независимо от количества ядер. Мы также лидируем в области виртуализации, со множеством открытых продуктов, использующих ее и конкурирующих за популярность. Это дает Linux неоспоримое преимущество на арене рабочих станций, которые, в свою очередь, просачиваются на рынок продвинутых домашних пользователей. Наш ультрасовременный Linux-компьютер демонстрирует, где у Linux преимущества в наши дни и где Linux может стать еще лучше после легкой подстройки драйверов там и сям. LXF

Слово за вами!

Печальный урок, усвоенный нами при подготовке этой статьи, заключается в том, что поддержка драйверов в Linux для новейшего и крутейшего оборудования порой может отсутствовать. Конечно, можно воспользоваться Google и отыскать малопонятную примочку, с которой все работает, и конечно, многие новые комплектующие будут поддерживаться «в следующем релизе ядра», но достаточно ли этого? Следует ли любителям игр отказаться от Linux, или нам нужно нажимать на производителей оборудования, добиваясь лучшей поддержки драйверов для альтернативных ОС? Заслуживают ли двоичные драйверы внимания, или разработчики должны тратить больше времени на разработку настоящих открытых драйверов для всех наших устройств?

У вас есть шанс позволить миру услышать ваше мнение – пишите нам на letters@linuxformat.ru с темой «Ultimate Linux Hardware», и высказывайте свои доводы!

Что внутри?

По сравнению с предыдущим суперкомпьютером, в этом году увеличение цены на 500 фунтов стерлингов дало 4-х кратное увеличение скорости.

	Современный Linux-компьютер LXF53	Цена без НДС	Современный Linux-компьютер LXF92	Цена без НДС
Процессор	AMD Athlon 64 3400+ 2.2 ГГц, 1 Мб кэша 2 уровня	£247	Intel Core 2 Quad Extreme, 2.66GHz, 2MB кэша 2 уровня	£539.99
Материнская плата	MSI K8T Neo	£82	Abit IN9 32X-MAX	£178.99
Оперативная память	512 Мб 400 МГц (PC3200) Corsair XMS	£51.99	2 Гб 800 МГц (PC6400) Corsair Dominator	£139.99
Жесткий диск	250 Гб SATA Maxtor MaXLine Plus II	£131	750 Гб SATA II Seagate Barracuda	£142.21
Оптический привод	TDK DVD-RW/+RW (IDE)	£87.99	NEC DVD-RW/+RW/RAM/DL (SATA)	£17.79
Видеокарта	Gigabyte GeForce FX 5950 Ultra, 256 Мб RAM	£275	XFX GeForce 8800 GTX, 768 Мб RAM	£340.99
Звуковая карта	SoundBlaster Audigy 2 ZS	£135.17	SoundBlaster X-Fi	£25.49
Корпус	Coolermaster Praetorian	£111	Coolermaster Cavalier	£36.99
Кулер процессора	Coolermaster heatsink and fan	£8	Zalman CNPS9700	£31.59
Мышь и клавиатура	Microsoft Wireless Optical Desktop	£42.54	Microsoft Natural Value Pack	£19.99
Монитор	17” CTX PV722E, 1280x1024	£308.24	24” Dell 2407 FPW, 1,920x1200	£444.99
Общая стоимость		£1,662.92		£2,201.21

Закодируй Greenфон

Для разработки вашего первого мобильного приложения не надо ни быть крутым хакером, ни даже иметь сам Greenphone, пишет **Грэм Моррисон**.

Trolltech Greenphone – открытая платформа, и инструменты для разработки под нее доступны по лицензии GPL. А значит, еще никогда не было так просто создать собственное приложение для мобильного телефона – до того просто, что нам хватит трех страниц, чтобы показать это. Наша программа будет всегонавсего пересчитывать сантиметры в дюймы, но ее так же легко можно употребить как конвертор валют или расширить любыми способами. Знакомство с C++ вам поможет, но если вы таковым не располагаете – вот отличный шанс научиться.

Greenphone использует особую версию Trolletch *Qtopia* API, у которого много общего с Qt API, используемом для создания KDE-приложений. Community SDK от сообщества Greenphone (**www.qtopia.net/modules/mydownloads/singlefile.php?lid=22**), необходимый для разработки GPL-приложений для Greenphone, представляет собой образ CD. Он построен на основе *VMware* для использования вместе со свободно распространяемым *VMware Player*. Это предустановленный дистрибутив Linux, включающий набор инструментов разработчика GNU, а также собственные библиотеки Trolltech, заголовочные файлы и приложения с графическим пользовательским интерфейсом. Так как это образ *VMware*, то вам не нужен Linux – *VMware Player* также доступен и под Windows. Но, что более важно, вам не нужен даже Greenphone: Trolltech предоставляет собственный «Эмулятор Greenphone», и вы можете играть с виртуальным телефоном, не боясь повредить оборудование.

Создание среды разработки

Первый шаг – установка Community SDK со скачанного образа CD. После этого у вас появится каталог с файлами *VMware*, которые необходимо загрузить в виртуальную машину. После ее запуска вы увидите зеленые тона экрана рабочего стола Trolltech KDE. На нем разбросано несколько иконок, и две из них используются для эмуляции среды Greenphone. Нажмите на иконке Runqvfb, затем на иконке Runqpe.

Комплект разработчика Greenphone использует образ *VMWare*. И бесплатный *VMware Player*..

Первая иконка запустит виртуальный фрейм-буфер для Greenphone, встроенный в графическое представление телефона. Вторая иконка запустит ОС *Qtopia* внутри эмулируемой среды [строго говоря, *Qtopia* не является самостоятельной ОС – она базируется на Linux, – прим. ред.]. В результате вы увидите виртуальное представление Greenphone на рабочем столе, которое ведет себя как настоящее, когда вы нажимаете на кнопки. (Любители повозиться с оборудованием могут подробнее узнать о Greenphone на сайте **www.trolltech.com/products/Qtopia/greenphone** или в обзоре LXF91.)

Виртуальная машина включает каркас для простого приложения

Greenphone. Вы можете обнаружить его, нажав на иконку Konsole для открытия командной строки и набрав **cd projects/application** для перехода в каталог, содержащий исходный код. Просмотрев содержимое каталога, вы увидите исходный код (**example.h** и **example.cpp**), пользовательский интерфейс (**examplebase.h**) и документацию в каталоге **Help**. Только об этих файлах и надо позаботиться при создании собственного приложения.

В командной строке наберите source **/opt/Qtopia/SDK/scripts/devel-x86.sh**, чтобы установить переменные окружения для нормальной компиляции примера. Затем потребуется команда **qtopiamake**, она подготовит исходные файлы примера для механизма сборки *Qtopia*. После этого пробежитесь по знакомым командам make clean, **make** и **make install**, чтобы собрать и установить пример проекта. После этого необходимо попросить Greenphone обновить свой список приложений, так что наберите **gph -rescan**.

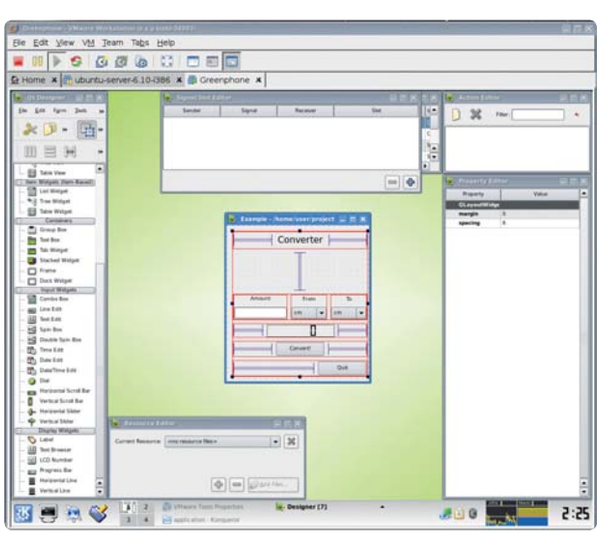
Теперь вы готовы запустить полученное приложение, как с помощью виртуального Greenphone, так и настоящего оборудования, если вы его подключили. Просто перейдите в папку Приложения (Applications) в телефоне и выберите иконку Example (она выглядит как красный знак вопроса на белом фоне). Запустится простое приложение, которое Trolltech включило в состав SDK – и вы сразу же обнаружите, что оно не делает ничего. Абсолютно ничего. Все, что можно сделать, это нажать на Выход (Quit). Наша задача – заставить программу выполнять полезную работу.

Проектируем пользовательский интерфейс

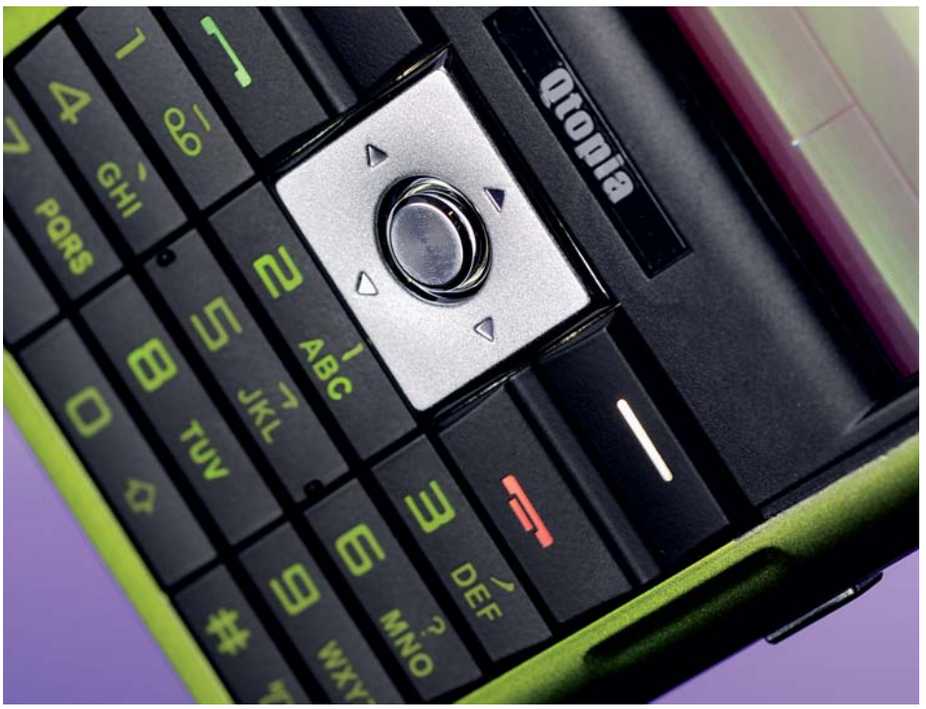
Прежде чем добавлять какую-либо функциональность, необходимо спроектировать пользовательский интерфейс. Как и для других KDE- и Qt-приложений, для этого есть инструмент *Designer*, и вы должны видеть его иконку на рабочем столе – она похожа на пару штангенциркулей. При первом запуске программа спросит, какой вид формы вы хотите спроектировать. Так как мы собираемся загрузить пользовательский интерфейс из каталога примеров Greenphone, то нажмите Закрыть (Close).

Если вы никогда не видели *Designer*, то, возможно, его интерфейс вас слегка отпугнёт. Панели разбросаны по всему экрану – он выглядит как *Gimp* на стероидах. Каждая панель заполнена опциями: небольшими окошками и выпадающими меню. Вам надо отыскать опцию Open Form (Открыть форму) в меню File (Файл). Далее выберите Projects (Проекты) > Application (Приложение), и откройте затаившийся там файл **examplebase.ui**. Теперь вы должны узреть интерфейс нашего пустого приложения для Greenphone.

Сейчас нам необходимо создать интерфейс пользователя для программы-конвертера. Нажмите на текст внутри окна, затем щелкните правой кнопкой и выберите Break Layout (Разбить компоновку) из меню Form (Форма). Таким образом мы разблокируем компоненты внутри окна и сможем передвигать их и добавлять новые. Нам понадобятся одно поле ввода, которое будет принимать входную величину, и два выпадающих меню для выбора входного и выходного формата. Также потребуется поле вывода для отображения конечного результата и кнопка для осуществления преобразования.



› **Экспериментируя с пользовательским интерфейсом в программе *Trolltech Designer*, найдите наилучшей вид для вашего приложения.**



› **Чтобы ввести числа в *Qtopia*, полсекунды подержите клавишу нажатой.**

Чтобы добавить эти компоненты, перетащите их из панели компонентов, расположенной слева на экране. Из раздела Display Widgets нам понадобятся четыре метки и один LCD Number для вывода результата. Из раздела Input Widgets выберите два Combo Box и один компонент Line Edit для ввода значения. Наконец, нам нужна кнопка Push Button из списка Buttons.

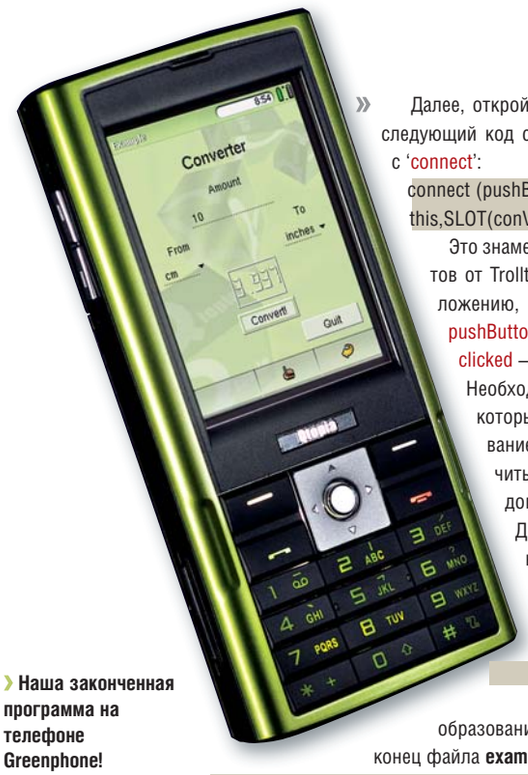
Выделите исходный текст примера и нажмите Delete, чтобы удалить его с формы – мы оставим только кнопку Quit. Можете расположить компоненты по своему вкусу. Текст на трех метках изменим на Amount, From и To соответственно (щелкните правой кнопкой мыши и выберите Change text (Сменить текст)). Если вы хотите выровнять текст по центру, то выберите свой компонент и свойство alignment из Property Editor (Редактор Свойств) и измените горизонтальное выравнивание на AlignHCenter.

Последнюю метку можно использовать как заголовок для приложения. Размер шрифта для этого компонента может быть выбран из списка шрифтов в редакторе свойств. Вам также потребуется изменить текст на Push Button, и добавить пункты преобразования в Combo Box'ы (щелкните правой кнопкой и выберите Edit items (Редактировать элементы)). Для начала добавьте inches и cm.

Добавив и отредактировав все необходимые компоненты, пробегитесь по руководству расположения компонентов на следующей странице, затем сохраните результаты, переписав оригинальные файлы. Если у вас не хватает прав это сделать, то откройте *Konsole* и наберите **sudo chmod -R 777 projects**.

Пишем код

Пользовательский интерфейс спроектирован; время добавлять функции, чтобы конвертор обрел законченную форму. Мы можем это сделать, добавив одну функцию в наш пример. Откройте файл **example.h** и добавьте 'void conVert();' сразу под методом 'void goodbye();' в разделе 'private slots'. Именно эта функция будет выполняться, когда вы нажмёте на кнопку Convert (Преобразовать) в приложении. »



Наша законченная программа на телефоне Greenphone!

Далее, откройте файл **example.cpp** и добавьте следующий код сразу за строкой, начинающейся с 'connect':

```
connect (pushButton,SIGNAL(clicked()),
this,SLOT(conVert()));
```

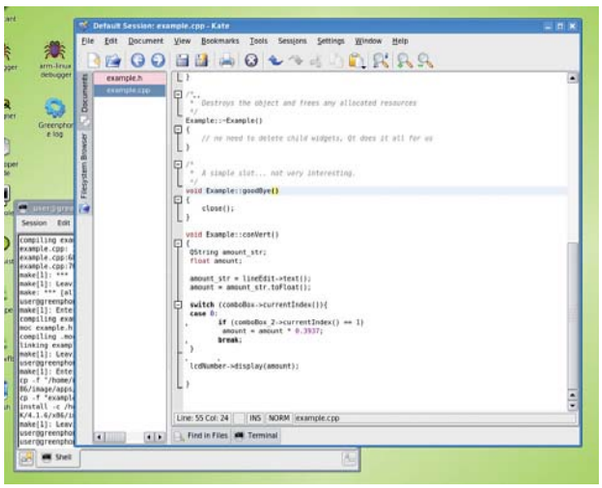
Это знаменитый механизм сигналов и слотов от Trolltech, и мы здесь сообщаем приложению, чтобы при нажатии на кнопку pushButton выполнялась функция conVert: clicked – это сигнал, а conVert – это слот. Необходимо добавить код для слота, который будет выполнять преобразование, а для этого нужно подключить заголовочные файлы для каждого используемого компонента. Добавьте в начало файла следующий код:

```
#include <qstring.h>
#include <qlineedit.h>
#include <qlcdnumber.h>
#include <qcombobox.h>
```

Наконец, добавим функцию преобразования – припишем следующий код в конец файла **example.cpp**:

```
void Example::conVert()
{
    QString amount_str = lineEdit->text();
    float amount = amount_str.toFloat();
    switch (comboBox->currentIndex()){
        case 0:
            if (comboBox_2->currentIndex()) == 1)
            amount = amount * 0.3937;
            break;
    }
    lcdNumber->display(amount);
}
```

Наберите **make**, **make install** и **gph -rescan**, чтобы установить вашу новую программу, и запустите ее из списка приложений (она все еще называется Example). Вкратце, мы делаем вот что: считываем значение из поля ввода и преобразуем его в число, а потом проверяем значения в обоих выпадающих списках. Мы предполагаем, что первый вход будет cm, а второй – inches. Если cm выбрано в качестве первой записи, то выражение if проверит значение во втором Combo box. Если это '1', то это должно быть, дюймы, и мы производим преобразование величины, полученной из поля ввода. Наконец, мы посылаем это значение на LCD-дисплей приложения.

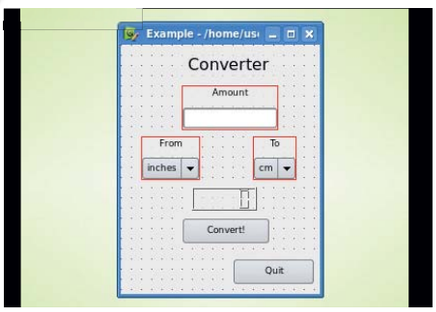


В текстовом редакторе *Kate* удобно редактировать исходный код: подсветка синтаксиса облегчает работу.

Очевидно, надо еще немало потрудиться, прежде чем программа станет действительно полезной, но она функционирует. И к тому же занимает меньше десяти строк! Вот краткий список возможных нововведений. Добавьте другие возможности конвертации – дюймы в сантиметры, например, или метры в ярды. Потом проверьте корректность ввода и отобразите ошибку, если введенное значение не есть число. Вы даже можете добавить возможность сохранять наиболее популярные преобразования.

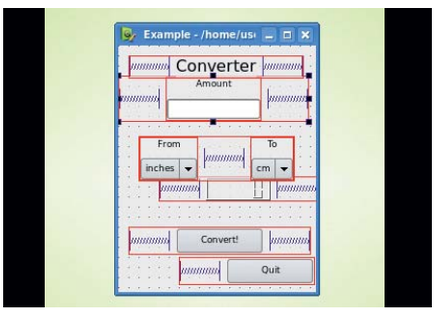
Если вы закончите этот проект, мы будем рады увидеть результаты. Пошлите нам ваш исходный код (letters@linuxformat.ru), и мы положим его на наш DVD. LXF

Шаг за шагом: Располагаем компоненты в Designer



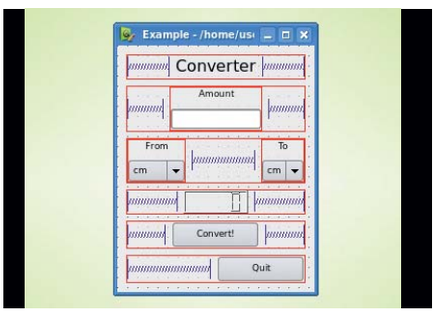
1 Группируем компоненты

Выбрав все компоненты для вашего приложения в *Designer*, расположите их, как вам нравится. Далее нажмите Shift и выберите каждый компонент с меткой, потом выберите Layout Vertically из меню Form. В результате создастся группа компонентов с менеджером компоновки, который будет видеть их как один компонент.



2 Выравниваем

Designer использует набор взвешенных 'разделителей' для форматирования раскладки. Перетащите горизонтальный разделитель из палитры инструментов в пространство между каждым компонентом, располагающимся в каждой строке. Далее выберите все разделители и компоненты в каждой строке и выберите Layout Horizontally (Расположить горизонтально) из меню Form.



3 Делаем сетку

Собрав все компоненты в группы и расположив их горизонтально по строкам, выберите Lay Out In A Grid из меню Form. Так вы добавите ваши группы компонентов в матрицу, которая будет масштабироваться в зависимости от размера и разрешения используемого экрана. Если вы захотите что-то поменять, отмените компоновку – выберите Break Layout.

LINUX FORMAT

Учебники

Наши эксперты помогут вам с любым приложением Linux

ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Свободный код свободному космосу

«Моритури Нолумус Мори» или Идущие на смерть этого совсем не приветствуют.»

Эмблема первых комопроходцев Плоского мира.

Терри Пратчетт. Последний герой.

Можно долго дискутировать на тему того, что или кто тащит упирающееся и визжащее человечество вперёд по дороге прогресса. Определённо, не деньги и уж тем более, не реклама. А вот крупные структуры, занимающиеся фундаментальными проблемы и финансируемые на деньги налогоплательщиков, время от времени выдают интересные идеи и результаты.

Не хотите вложиться в космическую экспансию человечества? Если да, то открытый проект CosmosCode (http://www.cosmoscode.org/) от NASA возможно то, что Вы ищете.

NASA – это большая и довольно богатая структура, в которой работают уникальные специалисты. Зачем им допускать сообщество к разработке программного обеспечения для космических миссий? Что хорошего могут «накодить» любители?

Собственно говоря, почему именно любители? Именно по той причине, что NASA является большой структурой, обмен между различными её частями чрезвычайно затруднён и открытый код вполне – естественный выход за пределы внутренних барьеров. Ну и кто сказал, что специалист обязан быть суперпрограммистом? Специалисты хороши в том, что они знают, что делать – это стратегия. А как делать – это уже тактика и помощь в этом вполне может оказаться весьма и весьма кстати. Кроме того, критических программ, которые, собственно говоря, управляют полётом не так уж то и много и вряд ли их удастся потом использовать повторно. Но есть ещё горы сопутствующего ПО, которого всегда много. Решая действительно сложные проблемы, люди выдают массу интересных вещей, но далеко не всегда их доделывают, так как действительно сложные вещи времени на это не оставляют. Просто безмерное поле для деятельности. Так что свободное программное обеспечение вполне вписывается в открытый космос.

Да, Debian 4.0 Etch вышел. Не совсем в тему, зато приятно.

E.M.Baldin@inp.nsk.su

В этом выпуске...

54 Неприкосновенный резерв
Медлительность – вор времени, она же – вор, часов, дней и месяцев работы. **Энди Ченнел** научит не откладывать создание резервных копий в долгий ящик

62 SugarCRM на свой лад
Вы все подстроили и отрихтовали – самое время двигаться дальше. **Марк Бейн** завершает серию статей о SugarCRM рассказом о добавлении своего собственного функционала.

58 Мониторинг устройств
Вы уже прочитали спецрепортаж этого номера и собрали компьютер своей мечты? Позвольте **Грэму Моррисону** показать, как защититься от взрыва, вызванного остановкой одного маленького вентилятора!

66 Теория ООП
Пол Хадсон на время забыл о практике и ударился в теорию – теорию объектно-ориентированного программирования. Она будет вам полезна – честное слово.

70 Управляя Java
Крутые Java-разработчики используют крутые штуки, чтобы управляться с исходными текстами и компилировать из них крутые программы. Обо всем этом и расскажет **Скотт Дуглас.**

74 Первые шаги в царстве гномов
Задумали помочь сообществу не словом, а делом? **Андрей Боровский** готов показать вам, как пишутся настоящие GNOME-приложения.

78 Ncurses-II
Армия фанатов Ratpoison имеет право воспылать справедливым гневом – **Андрей Боровский** говорит, что мышь бывает полезна даже в консоли, и подкрепляет свои слова примерами на языке C!

82 «Модель-вид-контроллер»
Теория такого уровня не снилась даже Полу Хадсону! **Антон Черноусов** расшифровывает слово «паттерн» и показывает, как применять его для написания практичных программ Java EE.

86 «Алло, справочная?»
Книга без оглавления – как небоскреб без лифта: жить можно, но неудобно. **Евгений Балдин** раскрывает всю силу справочного аппарата издания.

91 Chroma Key
Петр Семилетов наводняет города динозаврами, чтобы показать нам всю волшебную силу Cinelerra!

Совет месяца: В обьятиях Берил

Мы уже попривыкли к Compiz и Beryl, и прошедшие полгода показывают, что существуют некоторые настройки и опции, способные здорово влиять на производительность и стабильность. Хотя Compiz и Beryl – [пока еще] разные проекты, оба они базируются на одном и том же коде, так что наши советы будут полезны всем.

Самый сильный удар по производительности наносит опция Sync To VBlank, которую можно найти во вкладке General Options. Она включена по умолчанию, а ее суть – обеспечить, чтобы 3D-эффекты, рассчитанные Beryl или Compiz, отображались ровно в тот момент, когда экран физически обновляется. Но есть и обратная сторона медали: синхронизация с вертикальной разверткой вашего монитора может привести к тому, что отрисовка будет выполняться медленнее, чем могла бы. Если вы работаете не на суперкомпьютере, эту опцию следует выключить.

Второй «тормоз» – это эффект размывания (blur). Следует признать, что выглядит он впечатляюще – смягчает фон за заголовком окна или обеспечивает эффект «смазывания при движении», но все это не стоит потерянной производительности. Гораздо лучше отключить его и наслаждаться реактивным рабочим столом. Если у вас почти приличная видеокарта (а уж раз вы решили работать с Compiz или Beryl, так и должно быть), включите мип-мэппинг для куба рабочего стола. Это означает использование высококачественных, аппаратно масштабируемых текстур, а не грубого приближения, что используется по умолчанию.

Если вы используете Beryl и хотите узнать, насколько все стало быстрее, попробуйте модуль Benchmark со страницы Extras. Он открывает небольшое окно, в котором показывает, увеличивают или уменьшают используемые вами эффекты среднее число кадров в секунду. Наконец, если вам действительно нужна наилучшая производительность и максимальная гибкость, попробуйте перейти с Xgl на AIGLX. Он лучше поддерживается и

Система: Политика

Резервные копии, как пластырь: лежат себе смирно, пока вы не порежетесь. **Энди Ченнел** научит вас, как всегда иметь в запасе такую «аптечку».

Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется еще со времен Dragon 32

В начале была создана Вселенная. Вообще это расценивается как неверный ход: он обусловил провал в 15 миллиардов лет до появления домашнего ПК, сделавшего ‘жизнь’ более продуктивной. Прошло 15 миллиардов лет и один день с Большого взрыва, прежде чем Энди Ченнел наконец затеял резервирование **/home/Documents/Work**, и очень неудачно, потому что тремя часами ранее многие жизненно важные для него документы безвозвратно исчезли в темной материи кибер-пространства. А он и не подозревал о трагедии – пока, спустя еще шесть месяцев, коллеги не попросили его прислать кое-какие из этих файлов.

Любой постоянный пользователь компьютера сталкивался с паникой восстановления, и часто только трагическая ситуация подвигает нас на реальное создание и использование политики резервирования. Этот учебник поможет вам создать разумную (и достижимую) политику до того, как трагедия произошла. Мы рассмотрим, что следует резервировать в системе и документах, как использовать инструменты популярных приложений для должного создания копий ваших ценных документов и как убедиться, что вы зарезервировали именно то, что хотели, и в будущем сможете это восстановить.

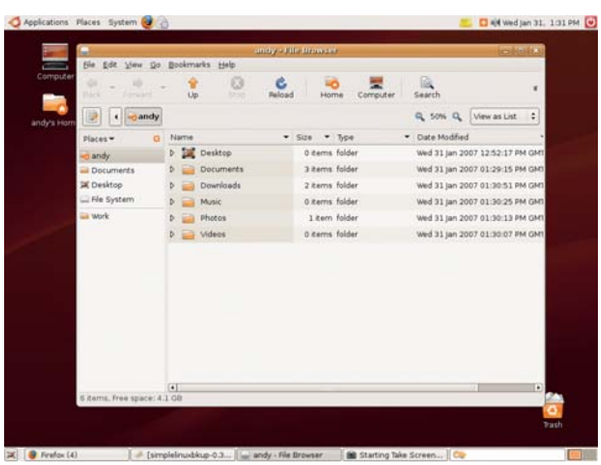
Имеется немало прекрасных Linux-приложений для резервирования, например, *Amanda* и *Mondo*, но для нашего варианта резервирования они слишком сильнодействующие (ибо созданы с прицелом на системных администраторов). Поэтому мы будем использовать приложение под названием *Simple Linux Backup* – оно базируется на Java и не зависит от дистрибутива, и, как следует из названия, просто в использовании. Мы вернемся к *Simple Linux Backup* в разделе Как это сделать на стр. 56. Но сначала выявим те части системы, которые нуждаются в резервировании.

Жизненные показания

Некоторые главные вещи в ПК важнее других , и для среднего домашнего пользователя это личные данные. Они включают все, от фотографий и видео до документов *OpenOffice.org*, электронных писем и звуковых файлов. Расстановка приоритетов удобна тем, что хотя эти файлы могут быть большими – или очень большими, в случае видео – все они имеют свойство собираться в одном месте, а именно, в вашем домашнем каталоге, и это весьма упрощает их резервирование. Конечно, файлы могут быть и раскиданы повсюду, так что первая задача при построении политики резервирования – создать грамотную структуру файлов (и жестко ее придерживаться).

Большинство дистрибутивов предоставляют внутри **/home/имя-пользователя** каталог **Documents**, и в него я хочу добавить квартет новых папок: **Photos**, **Videos**, **Music** и **Downloads** – обратите внимание на использование заглавных букв и очевидность категорий. Это полезно не только для резервирования, но также облегчает повседневную навигацию и управление, поскольку легко запомнить, что все загружаемое из Интернета помещается в **Downloads**, изображения с вашей фотокамеры – в **Photos**, и т.д. [не следует называть папки кириллицей: из-за этого бывают проблемы, – прим.пер.] Это также удобно для настройки расположения файлов по умолчанию в приложениях вроде *Amarok* или *F-Spot*: кому охота искать повсюду каталог с непонятным именем, которое казалось таким милым после семи стаканов.

Не все каталоги надо резервировать одинаково часто. Например, незачем еженедельно сохранять файлы, извлеченные из вашей CD-коллекции – не только потому, что этот каталог вряд ли меняется на каждом шагу, но и потому, что резервная копия и так красуется в



› Разумный и удачно названный набор каталогов позволит сократить время поиска в резервных копиях.

» **Месяц назад** Безопасность и практическая выгода от прав доступа и паролей.

резервирования

алфавитном порядке на вашей полке CD, а то и на MP3-проигрывателе. Как правило, у меня есть одна копия моего каталога **Music**, на нескольких DVD, а также на второй машине и на 40-ГБ iPod'е.

Точно так же, **Photos** и **Videos** нуждаются в регулярном резервировании только если вы постоянно пополняете каталог. Однако мультимедийные файлы нередко не поддаются восстановлению после несчастного случая, так что некоторая паранойя тут не помешает. Видео не является особой проблемой, поскольку поступает (обычно) с DV-пленки, которой от процесса перекодирования не убудет, и обычно предназначено для DVD, поэтому достаточно будет создать пару физических копий, протестировать их и проверить, а потом нормально зарезервировать перекодированные файлы и спокойно их удалить.

С фотографиями по ряду причин проблем может быть чуть больше. Во-первых, большинство людей чаще снимают фото, чем видеосюжеты, а значит, эта папка пополняется тоже чаще; во-вторых, карты памяти в фотоаппаратах нередко очищаются, иногда прямо в процессе переноса файлов, так что резервной копии не остается; и, наконец, неэффективно записывать CD при каждом пополнении своей коллекции, если добавились какие-то 10 МБ.

Вследствие этого я разбил процесс резервирования фотографий на стадии. С фотоаппарата изображения импортируются в *F-Spot* и удаляются из камеры. Затем они копируются в специальный каталог на другой машине (flash-диск тоже подойдет). Когда этот каталог разрастается до 600 МБ, я проверяю файлы, чтобы убедиться, что все они на месте и не испорчены, а затем записываю CD, помечая его датами первого и последнего снимков. Затем – уж такой я параноик – я передаю этот CD родственникам: если мой дом сгорит, мне останутся хотя бы фотографии. А поскольку я самый-самый настоящий параноик, то еще и выгружаю наиболее важные из них на *Flickr* с пометкой Private. Так что если и мой дом, и дом моих родичей будут сметены с лица Земли, я все-таки сохраню часть своих драгоценных воспоминаний. Наверное, это уже перебор...

Такой тип резервирования – размещение информации в каталоге с последующей ее записью на CD – вполне пригоден, но простоват, и не очень-то подходит для частого резервирования, которого требует содержимое вашего каталога **Documents** и персональные системные настройки: электронной почты, *OpenOffice.org* и данных web. Прежде чем перейти к автоматическому процессу резервирования, рассмотрим плюсы и минусы различных медиа-носителей.

Где разместить?

Резервные копии надежны настолько, насколько надежны носители, на которых вы их храните, поэтому действительно важно убедиться, что вы выбрали правильное хранилище для ваших данных. Наиболее очевидный выбор – записываемые оптические носители, типа CD или DVD. Их плюсы – чрезвычайная дешевизна, портативность и, в случае DVD, большая вместимость. Технология зрелая и потому хорошо поддерживается в Linux.

Однако хранение на дисках не лишено проблем. Наибольшая трудность состоит в том, что DVD не перезаписываются (да, есть и DVD-RW, но только безумцы пользуются ими для резервирования!), и потому прекрасны для любых типов резервирования, кроме интересующего нас инкрементного. Они также не работают напрямую с используемыми здесь программами – хотя имеются приложения резервирования, например, *KDar*, отлично работающие с CD и DVD.

Метод без проблем – это резервирование на flash-диск или на внешний жесткий диск: оба они способны читать и писать с утра до вечера, причем гораздо быстрее, чем CD. Правда, при недолгом сроке эксплуатации он обойдется дороже (за длительный срок вы сэкономите на DVD-матрицах), да и портативность, в случае внешнего жесткого диска, пострадает.

Вариацией данного способа является резервирование на сетевой диск. Проблема домашних пользователей в том, что эта вторая машина скорее всего будет стоять недалеко от первой, так что риск от наводнения, пожара или кражи не снижается.

Последний метод – online-резервирование, при котором вы отсылаете резервную копию электронным письмом самим себе. И недорого, и безопасно: сервер, по крайней мере, находится не в том же здании, что и оригинальные файлы. Хотя имеются потенциальные проблемы кражи данных. Если вы используете online-резервирование, неплохо будет шифровать и защищать паролем пересылаемые по сети архивы. Правда, метод электронной почты имеет ограничение: большинство почтовых серверов оговаривают максимальный размер пересылаемого или принимаемого вложения. Обычно это 10 МБ – на сотню документов хватит, но отнюдь не спасет, если надо сохранить 3 000 фотографий.

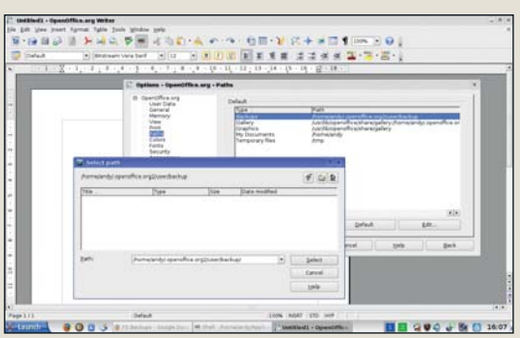
Если вам необходимо больше места, одним из недоиспользованных ресурсов является свободное web-пространство, весьма часто сопутствующее учетной записи у провайдера. Но будьте осторожны »

Скорая помощь

Настройте календарь так, чтобы он напоминал о 'времени резервирования' в пятницу вечером. Причем напоминал до скончания веков

Авторезервирование в OpenOffice.org

Похоже, что по мере роста сложности и надежности приложений, все больше людей для львиной доли своей работы будет использовать online-приложения вроде *Zoho Writer*, *ThinkFree Office* или *Google Docs*. В этой ситуации резервирование становится излишним, поскольку приложение сохраняет копию каждую пару секунд. Однако покамест все делается в *OpenOffice.org*, так что мудро было бы знать, как настроить пакет на автоматическое запоминание ваших писем, электронных таблиц и презентаций в определенное место. Согласно методу, описанному в данном уроке, это должен быть некий каталог для записи на CD, отправке по e-mail или расположенный на внешнем диске.



› Для указания места хранения резервных копий OOo используйте раздел Path [Пути].

Вы можете настроить резервирование для всего *OOo* из любого его приложения. Я покажу, как сделать это в *Writer*. В открытом приложении выберите пункты меню Сервис > Параметры и щелкните на небольшой иконке (знак +) рядом с пунктом OpenOffice.org. В появившемся списке выберите пункт Пути, для открытия диалога путей. В этом списке есть пункт Резервные копии; выберите его, а затем нажмите кнопку Правка.

Теперь, используя браузер файлов, переместитесь в то место, где вы хотите хранить ваши резервные копии. Им может быть любой смонтированный диск.

После этого перейдите в раздел Загрузка/Сохранение и в окне Общие найдите пункт Всегда создавать резервные копии. Установите его и нажмите OK. Теперь при сохранении документа его резервная копия также будет сохраняться в месте, указанном вами на первом шаге. Магия! Подобные опции есть в *KOffice* и *AbiWord*; просто найдите их в меню настройки.

» при его использовании: оно рассматривается как общедоступнное, и даже зашифровав и защитив паролем резервные копии, я бы поостерегся помещать туда любую секретную или очень важную информацию. Однако оно вполне подходит для размещения фотографий, и вы должны иметь к нему доступ через FTP-клиент, например, *Filezilla*, *GFTP* или даже *Konqueror*. FTP-адрес и пароль должен выдать вам ваш провайдер. Чуть более безопасный вариант – обратиться к коммерческим пакетам web-хостинга, в Великобритании это стоит около 4.99 фунтов в месяц [в РФ расценки начинаются с тех же 300–500 рублей, – прим.ред.]. Большинство пакетов содержат опцию запрещения доступа к некоторым каталогам – идеально для резервных копий. И, разумеется, вы можете обратиться к своим файлам с любого компьютера, подключенного к Сети.

Ради простоты я буду создавать резервные копии в локальном каталоге (на рабочем столе). Это удобно для их записи на диск или отправки в удаленное хранилище (по электронной почте или через FTP). Если вы резервируете на сетевой, внешний или flash-диск, просто замените используемую мной точку монтирования на другую.

Скорая помощь

Не сохраняйте резервные копии на перезаписываемых CD или DVD. Они работают не так, как надо

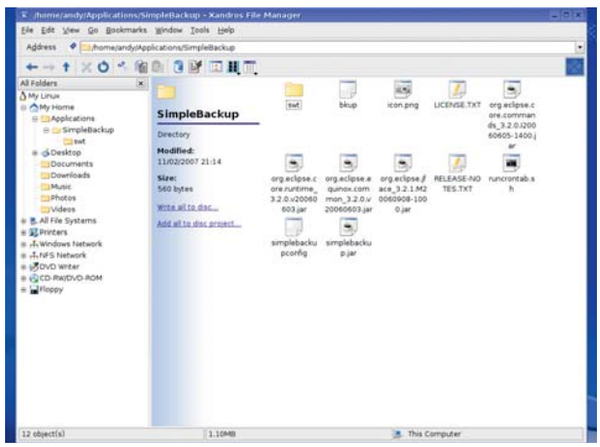
Как это делается

Программу *Simple Linux Backup* (**http://simplelinuxbkup.sourceforge.net**) написал и выпустил по лицензии GPL Стефан Розен [Stephen Rosen]. Это набор скриптов, увязанных Java и представленных в виде ясного графического интерфейса пользователя, использующего GTK. Список зависимостей – в большинстве дистрибутивов они удовлетворены – выглядит так: *Bash*, *tar*, *gzip*, Java, GTK и *cron*. Последний элемент не обязателен, но используется для настройки автоматического резервирования по расписанию, так что может очень пригодиться. У меня установлен Xandros, но сам метод будет одинаковым, независимо от используемого вами дистрибутива.

> «Скрипт будет брать файлы и складировать их в каталог Backup.»

Первая задача – заполучить программу. Загрузите последнюю версию с web-сайта и сохраните ее в вашем каталоге **/home**. Теперь откройте терминал (он находится либо в меню Аксессуары, либо в меню Система и должен называться Terminal, Konsole или Console) и наберите

```
tar xzf simplelinuxbkup-x.x.x-i386.tar.gz
```



› Наш пакет *Simple Linux Backup* даже не нужно устанавливать. Просто распакуйте его и запустите.

Эта строка просто распакует файл и создаст в текущем каталоге новый подкаталог с именем **SimpleBackup**. Часть x.x.x в предыдущей команде должна соответствовать номеру версии загружаемого приложения.

Теперь откройте *Konqueror* или *Nautilus*, найдите этот каталог и откройте его двойным щелчком. Само приложение *Simple Linux Backup* расположено среди набора Java-файлов (с расширением **.jar**) и имеет имя **simplebackupconfig**. Дважды щелкните на нем, чтобы запустить. Если вам предоставляется на выбор: запустить или запустить в терминале (так любит делать Ubuntu), выберите запуск в терминале. После этого откроется главная утилита настройки, определяющая, что следует резервировать, где разместить получившиеся файлы и когда следует выполнять резервирование.

Интерфейс пользователя утилиты настройки действительно прост. На левой панели – пустое пространство, готовое для заселения элементами, которые вы хотите резервировать. Выбор осуществляется среди каталогов или файлов; пункты, выбранные ошибочно удаляются кнопкой Remove [Удалить]. В правой панели можно определить элементы уже выбранных каталогов, которые мы хотим исключить из процесса резервирования. Например, если вы резервируете целый каталог музыки, но считаете, что вам ни к чему какие-нибудь заумные Throbbing Gristle, то можете выбрать для резервирования каталог Music, а затем нажать соответствующую кнопку, перейти к файлу или каталогу, которые хотите исключить, и выделить их обычным способом. Определившись с выбором, нажмите кнопку Next [Далее].

В следующем разделе производится настройка приложения на автомонтирование устройств резервирования (или, по крайней мере, на попытку такового) или на ожидание, пока вы смонтируете их вручную, что немаловажно, если вы регулярно делаете резервные копии на внешние носители, поскольку необходимо убедиться, что эти носители будут доступны в нужный момент. Здесь вы также окончательно настраиваете размещение резервных копий. Если вы резервируете локально (как я), а затем записываете данные на CD, вам следует создать каталог внутри домашнего каталога или на рабочем столе (назвав его как-нибудь вроде **backup**) и затем выбрать его. Если вы резервируете на USB Flash, сменный или сетевой диск, то вы, очевидно, должны указать здесь путь к нему. Кнопка Select... [Обзор...] выведет для этих целей стандартное окно выбора файлов.

После нажатия кнопки Далее вы перейдете к опциям управления и планирования. Первое весьма важно, если вы будете резервировать системные папки вроде **/etc**, содержащие пароли и т.п. По умолчанию, приложение резервирует от имени текущего активного пользователя, но если вы хотите иметь доступ к важным системным файлам, при-

Местоположение — это все!

В отличие от ключевых документов и медиафайлов, электронные письма и данные браузера сходу не найдешь. Ниже приведен список папок, которые вам необходимо резервировать, чтобы сохранить электронную почту, если с ПК приключится беда.

В большинстве случаев указанные файлы будут размещаться в вашем каталоге **/home/имя-пользователя** (по-научному, они записываются с **~/** в начале). Те из них, чье имя начинается с точки, являются скрытыми, так что вам нужно установить опцию Показывать скрытые файлы в меню Вид, чтобы их видеть.

Thunderbird:
» .mozilla-thunderbird/xxx.default/Mail/Local Folders (где xxx – произвольный набор символов)

Evolution:
» .evolution
» .gconf/apps/evolution
» .gnome2_private/Evolution

Kmail:
» .Mail (or it may be ~/.mail)
» .kde/share/apps/kmail
» .kde/share/config/kmailrc

Если вы хотите резервировать закладки вашего браузера и историю посещенных сайтов для *Firefox*, то наилучший вариант – использовать расширение *Browser Sync* от Google, доступное на **http://labs.google.com**. Оно клонирует всю вашу систему *Firefox*, сохраняет на сервере Google (естественно, в зашифрованном виде) и затем восстанавливает в любом web-браузере с установленным расширением.

Browser Sync – превосходное решение для резервирования при использовании дуальной загрузки или нескольких ПК (независимо от ОС); правда, некоторые люди опасаются вверять сугубо личную информацию всемогущей поисковой компании. На сей случай имеется несколько мощных online-инструментов для закладок, включая вездесущие Delicious, Sitejot и Yahoo Bookmarks, и все они прекрасно интегрируются в *Firefox*.

дется запустить его от имени root. Однако мы резервируем только документы, и можем оставить эти опции пустыми – то есть разрешить резервирование кому угодно. Я также собираюсь выбрать Silent Mode [Молчаливый режим], означающий, что скрипт будет запущен в фоне и не будет выдавать никакой информации о ходе процесса.

Наконец, осталось настроить расписание, чтобы скрипт резервирования запускался в определенное время. Для этого имеется две опции: можно для запуска скрипта использовать приложения вроде *KCron* или *Gnome-Schedule*, но проще позволить утилите настройки сделать все самостоятельно. То есть весь процесс резервирования является замкнутым, и позднее вы сможете его перенастроить, запустив утилиту настройки и уже не думая о двух приложениях. Итак, нажимаем Schedule With SBCP [Расписание от SBCP] и вводим время запуска резервирования, в 24-часовом формате. Оно будет запомнено. Теперь нажмите Finish [Завершить] для завершения процесса.

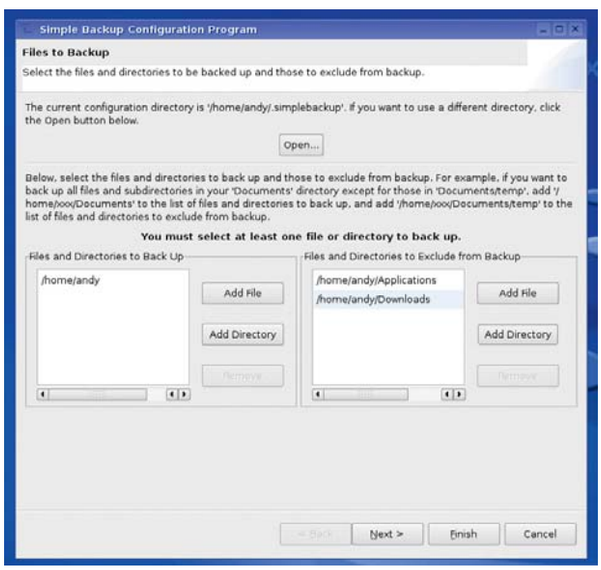
Понедельник – день особый

Далее произойдет следующее: когда настанет указанное для резервирование время, скрипт запустится (благодаря chrontab-системе Linux), соберет указанные файлы и каталоги и поместит их в каталог **Backup** на моем рабочем столе в виде tar-архива. Для экономии места приложение также сожмет архив при помощи gzip. Таким образом мы получим файл с именем **Backup.[day].tar.gz**, где **[day]** – день, когда скрипт был запущен в последний раз. Но это еще не все: Розен настроил скрипт так, что по понедельникам он запускает полное резервирование, то есть целиком переписывает указанные каталоги, а в остальные дни копирует только измененные файлы. Памятуя об этом, вы можете легко взять свой архив в пятницу вечером, записать его на CD или отослать на электронный ящик, и будете иметь полную копию недельной работы. Всем прочим управляет система, и вам остается только настроить календарь, чтобы он напоминал вам про это мероприятие.

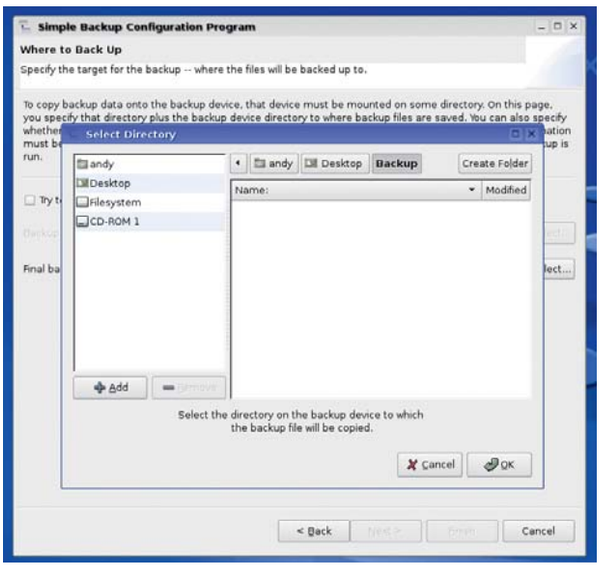
Представьте себе теперь, что катастрофа произошла, и вам необходимо восстановить последнюю резервную копию, сделанную *Simple Linux Backup*. Это производится тем же методом, который мы использовали при распаковке самого приложения после загрузки. Скопируйте файл резервной копии с вашего CD, USB-диска или online-хранилища в **/home/username**, а затем в терминале наберите

```
tar xzf Backup.[day].tar.gz
```

Команда распакует файл и восстановит все данные с соответствующей структурой файлов, созданной ранее, и мир снова станет прекрасным.



› **Simple Linux Backup совмещает скрипты с интуитивным GUI.**

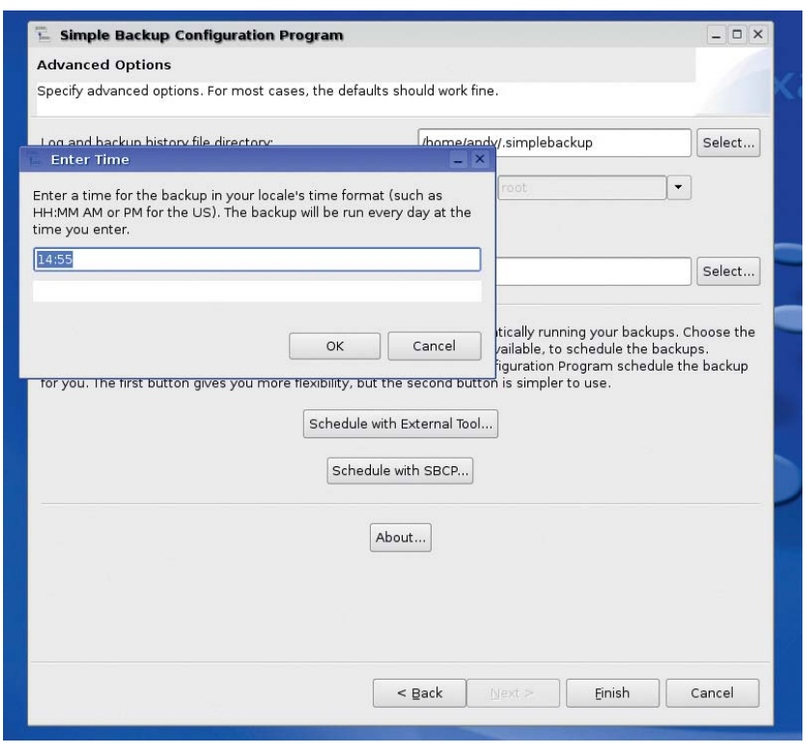


› **Выберите папку, в которую будете помещать накопительные резервные копии, и в конце недели скидывайте ее на CD.**

Quick tip

Once you've created a backup, check a selection of files to ensure they have been saved properly.

Если вы будете все это делать (тратя всего минут десять за неделю!), то в следующий раз, когда ваш ПК сгорит или взорвется, максимум, чего вы лишитесь, это недельной работы – а то и однодневной, если вы прилежны – а не нажитого непосильным трудом за 15 миллиардов лет. LXF



› **Приложение можно настроить на запуск резервирования в любое время дня.**

» **Через месяц** Где же C:? Разгадаем загадки файловой системы Linux.

Hardware: Следи за

Предупрежден – значит, вооружен. Слежение за устройствами – лучший способ предотвратить неожиданный крах системы, утверждает **Грэм Моррисон**.

Наш эксперт

Грэм Моррисон
Разработчик свободных программ, фанат KDE и энтузиаст электронной музыки. Он обсуждал синтезаторы не с одним ведущим членом сообщества FOSS.

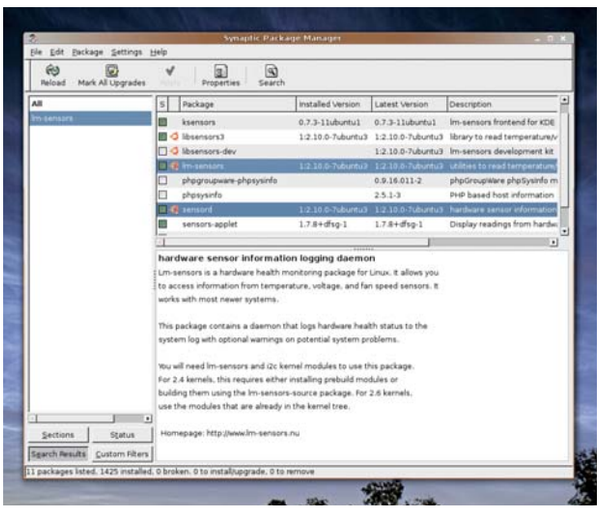
Важных причин для отслеживания состояния ваших устройств немало. К примеру, вы «разогнали» свою крутую Linux-машину и должны отслеживать температуру и напряжение на процессоре, чтобы не поджарить его; или втиснули все части компьютера в крошечный корпус, чтобы уместить его под телевизором, и хотите убедиться, что он не сгорит вместе со своим высоким экранным разрешением.

Большинство современных комплектующих, включая центральный процессор, жесткие диски, видеокарты и материнские платы, имеют один-два датчика, и слежение за их показаниями поможет вам создать более стабильную систему – ликвидируя проблемы до их превращения в отказ или утрату данных. Наблюдение за вентилятором позволит превратить замену дорогого устройства в покупку дешевого кулера. Но, пожалуй, самая веская из причин заняться слежкой за устройствами – ведь это же круто! На данном уроке я покажу, как применять продвинутые инструменты для контроля за датчиками вашей материнской платы и других устройств.

Мой первый датчик

За температурой процессора можно следить обычной командой. Дело в том, что модуль ядра, включенный во многие дистрибутивы Linux, умеет считывать показания прямо с чипа вашего процессора. Модуль называется *thermal*, и при загрузке он добавляет **/proc/acpi/thermal_zone** в файловую систему **/proc**. Наберите lsmod |grep thermal от имени системного администратора, чтобы узнать, загружен модуль или нет. Если да, то вы увидите его в списке вместе с модулем *processor*. Если нет, попытайтесь набрать modprobe thermal и загрузить его вручную. Теперь вы можете смерить температуру вашего процессора, набрав следующее:

```
cat /proc/acpi/thermal_zone/THRM/temperature
```



› ***Lm-sensors*** **имеется в менеджере пакетов большинства дистрибутивов; как и демон, протоколирующий показания в фоне.**

```
temperature:             40 C
```

[не во всех дистрибутивах; так, для Mandriva 2007.0 команда будет

```
cat /proc/acpi/thermal_zone/THM/temperature
```

прим. пер.]

Выведется температура по шкале Цельсия. С тем же успехом вы можете набрать acpi -t. Проблема обоих этих подходов в том, что они работают не на всех машинах: модуль *thermal* может не дружить с вашим оборудованием, или его поддержка может быть не встроена в ядро в вашем дистрибутиве. Другая проблема – таким образом вы узнаете только температуру процессора. А другая информация, вроде температуры материнской платы или насколько быстро вращается вентилятор для предотвращения перегрева, недоступна [это не совсем так – в каталоге **/proc/acpi** есть множество других файлов, доступных при загрузке соответствующих модулей ядра и содержащих, к примеру, степень заряда батареи ноутбука, – прим.ред.].

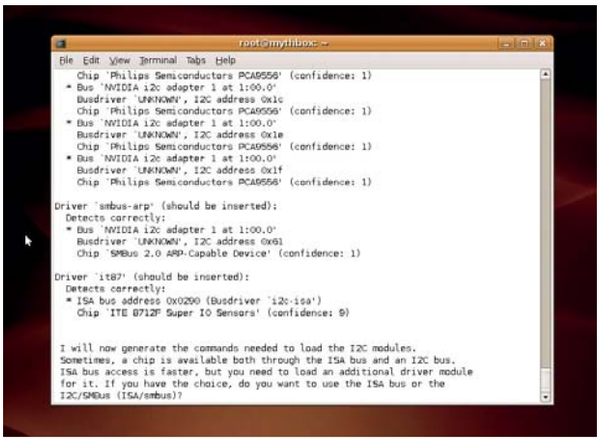
К счастью, есть решение получше, названное *Lm-sensors* – монолитный инструмент для слежения практически за всеми датчиками, имеющимися в вашей системе. Это раскидистый набор модулей ядра, файлов конфигурации, GUI для настройки и данных. Не пугайтесь трудных слов: *Lm-sensors* – настоящая нирвана мониторинга устройств.

Установка Lm-sensors

Ваш дистрибутив должен содержать *Lm-sensors* в виде установочного пакета. Просто откройте менеджер пакетов и поищите 'Lm-sensors'. Вы должны увидеть несколько пакетов, включая сам *Lm-sensors*, отдельную библиотеку (*libsensors3*) и фоновый демон (*sensord*), а также несколько графических оболочек для показа сведений, выдаваемых *Lm-sensors* в виде приборных досок и циферблатов. Но пока просто убедитесь, что необходимый пакет *Lm-sensors* установлен.

Все комбинации устройств предусмотреть невозможно, поэтому для работы *Lm-sensors* требует тщательной настройки. Он должен

работой системы



› ***sensors-detect*** **задаст столько вопросов, сколько обнаружит устройств.**

опросить различные компоненты устройств, проверить сигнатуры доступных датчиков ваших устройств и составить опись оборудования.

Инструмент, точно определяющий, что и где находится, называется *sensors-detect*. Он запускается из командной строки от имени администратора. Если при его первом запуске (хотя это маловероятно) произойдет ошибка, похожая на 'No i2c device files found', поищите на сайте *Lm-sensors* (**www.lm-sensors.org**) скрипт **mkdev.sh**. Он создаст недостающие ссылки в вашей файловой системе, потому что *Lm-sensors* сканирует устройства, только если видит ссылки на них.

Поисковый скрипт задаст пару вопросов, и практически все правильные ответы даны по умолчанию. Например, первый вопрос – хотите ли вы проверять I2C- или Smbus-адаптеры. Это первейшие кандидаты на информацию о датчиках, способные предоставить данные о потреблении энергии и состоянии батарей, если вы используете ноутбук, так что следует ответить «Да». Не беспокойтесь если при первом поиске не выявится ни одного датчика – второй проход выявит недостачу.

На современных компьютерах, весьма вероятно, датчики не обнаружатся, и программе поиска понадобится другой подход. При нем опрашиваются все подключенные к системе устройства, с целью определить, можно ли извлечь какие-либо данные о датчиках из их драйверов. Есть малый шанс краха системы при опросе плохо реализованного драйвера – не забудьте о нем, особенно если ваш свежесозданный в порыве вдохновения шедевр еще не сохранен. Именно по этой причине инструмент поиска датчиков информирует вас о каждом устройстве, которое тестируется в данный момент: если что-то пойдет не так, вы сможете пропустить проблематичное устройство при следующем запуске поискового скрипта. Но как правило, можно без опаски ответить «Да» на все вопросы. Типичный порядок опроса – сначала видеокарта, потом любые датчики на материнской плате, которые могут быть опрошены по шине ISA.

По окончании опроса устройств на предмет наличия датчиков, вам будет выдан список найденных датчиков, вместе с модулями, необходимыми *Lm-sensors* для чтения информации с них. Этот список важен, потому что фрагмент его необходимо перенести в файл конфигурации: там перечислены все драйвера ядра, которые необходимо загрузить до того, как можно будет получать данные с датчиков, и находится он между двумя маркерами #----cut here----. Этот фрагмент нужно вставить в конец вашего файла **/etc/modules**, чтобы драйвера загружались при запуске вашей машины.

Внесение изменений вполне можно доверить поисковому скрипту, но все же неплохо и проверить файл конфигурации на наличие ошибок. Если у вас не обнаружились заведомо присутствующие датчики, будет полезно обратиться к Интернету. Иногда утилита поиска не распознает ваши датчики, особенно если у вас новые устройства, и вы можете достигнуть большего успеха, скопировав и вставив чьи-то готовые настройки. Почти всегда находится кто-то, уже проторивший для вас дорогу.

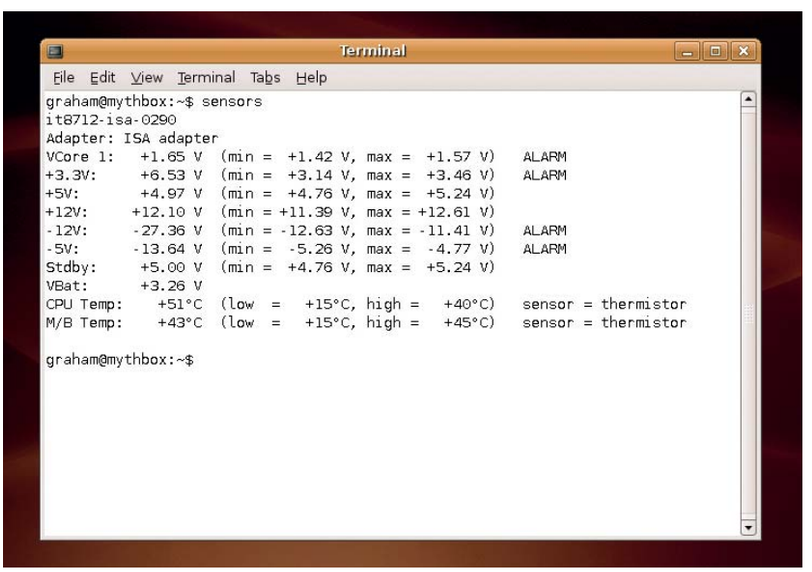
Калибровка

В моей тестовой системе по результатам поискового скрипта потребовалось загрузить только три модуля: *i2c-viapro*, *eeprom* и *it87*. Вы можете сделать это вручную (от имени администратора), набрав modprobe и затем имя модуля – то есть опробовать конфигурацию можно без перезагрузки машины. Простейшая команда опроса драйвера называется *sensors*, и выполнение ее от имени root обычно приводит к выводу одной или двух страниц (см. рисунок внизу). Вывод достаточно сырой: некалиброванный, часто без меток и с ошибками. Но его достаточно, чтобы понять: ваши датчики хоть что-то сообщают.

Файл конфигурации *Lm-sensors* находится в **/etc/sensors.conf**. Его следует отредактировать, если вы хотите сделать небрежный вывод команды *sensors* более точным и читаемым. Но прочесть его не так легко. Чтобы знать, что искать в этом файле, вы сначала должны определить проблемы, способные возникнуть с выводом команды *sensors*,. Например, температура моей материнской платы и процессора были перепутаны:

```
M/B Temp: +54°C (low = +15°C, high = +40°C) sensor = thermistor
CPU Temp: +41°C (low = +15°C, high = +45°C) sensor = thermistor
```

Чтобы найти соответствие меток и значений температуры, попытайтесь посмотреть в BIOS'е материнской платы. Большинство современных BIOS выводят список температур многих компонентов, и свя- »



› **При корректной настройке *Lm-sensors*, ввод 'sensors' снимет показания со всех аппаратных датчиков вашей системы: обычно на страницу или две.**

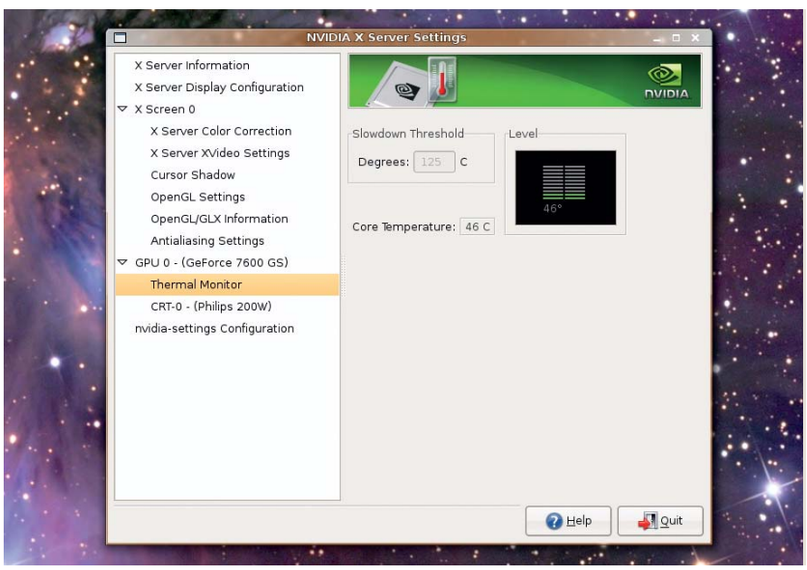
зать температуру от *sensors* с устройством легко, обнаружив то же число в BIOS. В моем примере BIOS сообщила температуру процессора 54°C, а материнской платы – 41°, ясно показав, что их значения переставлены местами. Вдобавок неправдоподобно, чтобы температура материнской платы была выше температуры процессора, если только плата не полыхает огнем.

Чтобы решить указанную проблему (я не про пожар!), найдите в файле конфигурации место, где датчику присваивается метка CPU Temp. Файл разбит на разделы для каждого поддерживаемого чипсета, и вам следует искать раздел, посвященный модулям и драйверам, которые вы добавили в **/etc/modules**. На моей машине это означает поиск раздела 'it87', а в нем поиск метки CPU Temp. Вот подходящие строки, а далее просто следует поменять местами метки temp1 и temp2:

```
label temp1 "M/B Temp"
label temp2 "CPU Temp"
```

Если имеются неверно определенные датчики, можете добавить к их записи 'ignore', тогда они не будут появляться в результатах, и точно также проигнорировать все неправильно считываемые вентиляторы и напряжения. Для показателя каждого датчика можно установить максимальное и минимальное значения, и при выходе за их пределы система выдаст предупреждение; но это проще настроить в графическом интерфейсе. Покончив с небольшими изменениями, необходимыми для улучшения вывода, сохраните файл и запустите программу sensors вновь, для проверки результата. Вы должны увидеть данные с каждого датчика с правильной меткой и калибровкой. Иногда верные значения можно подобрать только методом проб и ошибок.

Пользователи проприетарных драйверов от Nvidia, знайте: приложение nvidia-settings умеет следить за процессором видеокарты.



Если вместе с *Lm-sensors* вы установили демон *sensord*, то, настроив по своему вкусу *Lm-sensors*, перезапустите его. Воспользуйтесь инструментом управления службами вашего дистрибутива или наберите /etc/init.d/sensord restart. Теперь демон будет отслеживать любые неблагоприятные показания, которые можно найти в файле **/var/log/syslog**, а обнаруженное слово 'ALARM' укажет вам на дурное поведение данного устройства.

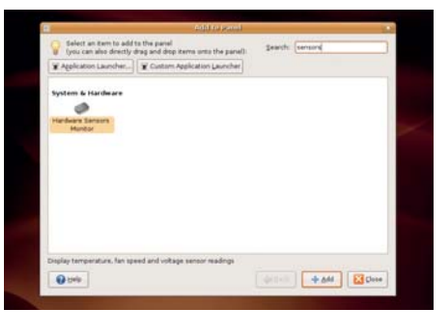
SMART жестких дисков

Рано или поздно ваш жесткий диск сломается – вот почему так важно резервировать ваши данные. Но жесткий диск редко ломается мгновенно. Часто в операциях чтения/записи диска число ошибок нарастает в течение нескольких дней или даже недель. ОС автоматически исправляет ошибки, оставляя вас в неведении относительно потенциальных проблем, таящихся в вашей Linux-машине. Поэтому жесткие диски получили индульгенцию на использование инструментов мониторинга, и большинство современных моделей имеют специальные датчики предупреждения неполадок, использующие систему под названием SMART – Self Monitoring, Analysis and Reporting Technology [Технология Самомониторинга, Анализа и Отчета].

Когда SMART пожалуется, что ваш диск сбоит, у вас должно быть достаточно времени на спасение всех важных данных, прежде чем он окончательно выйдет из строя. К сожалению, многие из нас отключают SMART в BIOS после известия, что ее включение приводит к небольшому понижению производительности. Но если вам дороги данные, хранящиеся на диске (а кому нет?), то небольшое снижение производительности – ничто по сравнению с потерями при полном отказе.

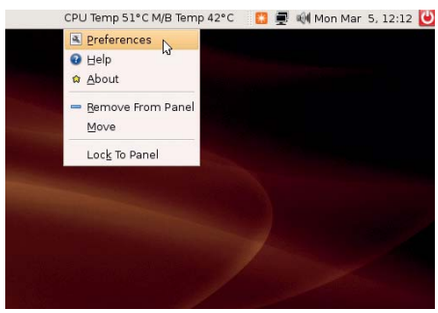
SMART хорошо поддерживается в Linux. Вам нужно установить пакет под названием *smartmontools* – в нем есть все необходимое для опроса вашего устройства и сообщения о любых проблемах. После установки пакета первым делом надо включить SMART на всех совместимых жестких дисках. Для IDE-дисков наберите smartctl -i /dev/hda, заменив hda файлом-устройством вашего жесткого диска; для

Шаг за шагом: Мониторинг датчиков в Gnome



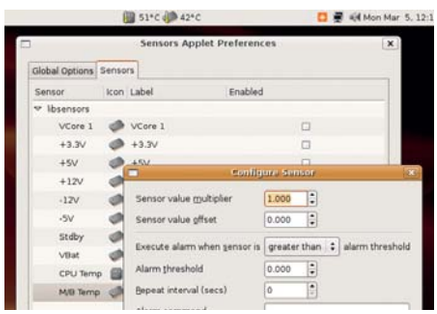
1 Добавим апплет

После установки панели датчиков при помощи вашего менеджера пакетов и настройки *Lm-sensors*, щелкните правой кнопкой на верхней панели Gnome и выберите Добавить на Панель. В разделе System & Hardware [Система & Оборудование] выберите Hardware Sensors Monitor [Монитор датчиков оборудования]. Апплет появится на панели.



2 Иконки по вкусу

По умолчанию апплет отображает температуру процессора и материнской платы в виде текста. Это можно изменить, щелкнув правой кнопкой на апплете и выбрав Preferences [Параметры]. Чтобы сменить текст на иконку, используйте опции на вкладке Global Options [Глобальные настройки].



3 Настроим датчики

Вкладка Sensors [Датчики] позволит выбрать иконку для каждого датчика, а также подключить другие датчики, настроенные в *Lm-sensors*. Также можно перекалибровать каждый датчик и настроить оповещение для случаев, когда что-то перегреется или треснет вентилятор.

SATA-дисков потребуется добавить -d ata в каждую используемую команду *smartctl*. Если вы не уверены, к какому узлу файловой системы подключено ваше устройство, посмотрите в файле **/etc/fstab**: там есть список устройств и их точек монтирования.

После ввода команды *smartctl* вы увидите некую информацию о вашем устройстве. Две строки будут начинаться со 'SMART': первая показывает, совместим ли ваш диск со SMART, а вторая сообщает, включен ли SMART. Если включен, можете расслабиться; если нет, введите smartctl -s on /dev/hda, чтобы исправить ситуацию.

Заведя на вашем диске SMART, разумно выполнить несколько тестов и убедиться, что он в хорошем состоянии. Наиболее простой тест – smartctl -H /dev/hda; в ответ устройство выдает текущий статус «здоровья». Сообщение 'Failed' означает, что велика вероятность кончины вашего диска в ближайшие 24 часа. Но даже если диск прошел базовый тест, не мешает подвергнуть его испытанию посерьезнее и убедиться, что все в порядке. SMART имеет мощный арсенал тестов для проверки работы вашего диска. Прежде всего это короткое самотестирование, по команде smartctl -t short /dev/hda. После запуска диск начнет самотестирование в фоновом режиме, сообщив приблизительное время завершения в командной строке. Обычно это занимает несколько минут, после чего команда smartctl -l selftest /dev/hda покажет последние девять результатов. Если все хорошо, вы увидите нечто вроде:

```
#1    Short offline    Completed without error    00%  17825    -
```

Первая колонка – номер теста, и SMART пытается сохранить как можно больше результатов тестирования (первыми выводятся самые недавние). Short offline – тип теста, а следующая колонка показывает отчет. 00% – это процент оставшегося теста (в данном случае нуль, поскольку тест завершен), а 17825 – время наработки диска в часах. Последняя колонка должна содержать первый блок с ошибками, обнаруженными в тесте. Если вы хотите самотестироваться по максимуму, попробуйте ввести smartctl -t long /dev/hda. Но знайте: потребуется 95 минут. Если вы не можете столько ждать, введите smartctl -X, и любое запущенное самотестирование прервется.

Естественно, несколько вручную запущенных тестов не способны исчерпывающе проанализировать работу вашего жесткого диска. Для более практически полезного мониторинга необходимо предусмотреть постоянное слежение за устройством и сообщения обо всех возникающих ошибках. Это работа демона *smartd*. Файл его конфигурации называется **/etc/smartd.conf**, и вам необходимо сначала отредактировать его, чтобы убедиться, что ваш диск правильно настроен. Синтаксис здесь тот же, что и для утилиты *smartctl*, и в файле содержится несколько типовых примеров, которые вы можете раскомментировать для себя.

Например, вот исчерпывающая запись для устройства hda:

```
/dev/hda -a -o on -S on -s (S/../../../02|L/../../6/03)
```

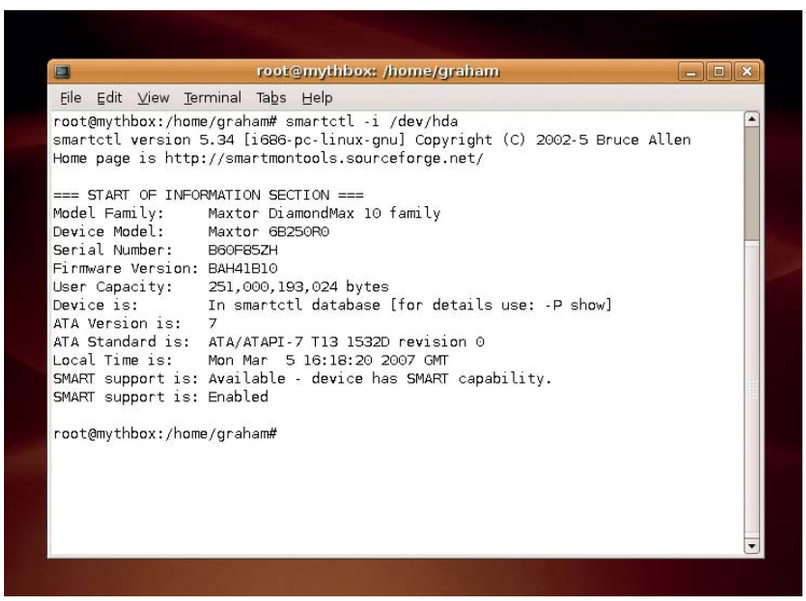
При этом будут отслеживаться все доступные атрибуты диска и запускаться короткое самотестирование еженощно с 2-х до 3-х и длинный тест каждую субботу с 3-х до 4-х часов ночи. Способ запуска демона зависит от вашего дистрибутива. Большинство систем запускают его автоматически после установки *smartctl*. Если это не так, можно обычным образом использовать панель настройки 'Службы' для запуска *smartd* или создать соответствующую ссылку в каталоге уровней запуска – наберите runlevel, затем посмотрите в **/etc/rc*.d**, заменив * вашим уровнем запуска. Любые проблемы автоматически зарегистрируются в вашем файле журнала **/var/ log/messages**, так что придется отслеживать и это – но вы же все равно всегда просматриваете журнал, правда?

Настроив вашу систему слежения за оборудованием, вы можете забыть о нем. И лишь в момент, когда что-то пойдет не так, вы поймете, что усилия, вложенные вами в настройку мониторинга, вознаграждены предупреждением о сбое до потери данных. Пусть даже ничего не сломается, но ведь круто! LXF

Скорая помощь

Вероятно, где-то кто-то имеет почти такое же оборудование, как у вас, так что шансы найти файл конфигурации Lm-sensors для вашего компьютера в Сети весьма велики. Сперва проверьте Lm-sensors wiki: www.lm-sensors.org/wiki.

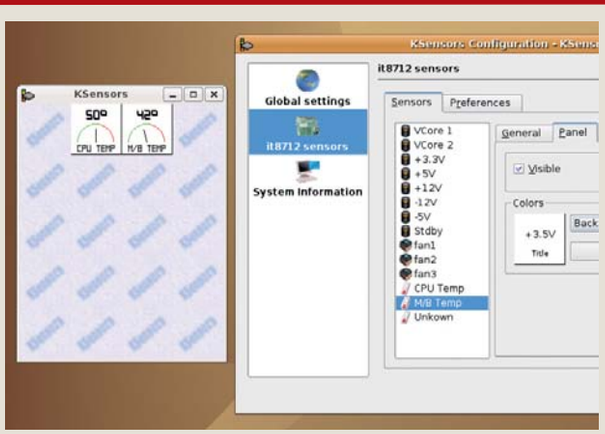
› Утилиты управления SMART сообщат о вашем жестком диске больше, чем прилагаемое к нему тощее руководство.



Создаем свою приборную панель с помощью виджетов

KSensors – прекрасный инструмент KDE для создания собственного приложения-монитора. *KSensors* использует *Lm-sensors* для получения необработанных данных, но позволяет привязать эти данные к выбранным вами виджетам и создать свою собственную приборную панель для имеющихся в вашей системе датчиков. Правый щелчок в исходно пустом окне выведет панель настройки со списком всех распознанных датчиков, и вы сможете выбрать любой из них для добавления на приборную панель или в системный лоток, точно определив его внешний вид. Среди виджетов – вольтметры, копии ЖК-панелей и термометры. Как и в других приложениях-«манометрах», можно перекалибровать значения каждого датчика, а также установить уровни оповещения.

Если вам кажется, что *KSensors* несколько перегружен, попробуйте *SuperKaramba*. Здесь имеется несколько встроенных виджетов для слежения за типовыми датчиками, размещаемых на фоне рабочего стола. Настроек не требуется, и места на рабочем столе требуется очень мало.



› *KSensors* – прекрасная графическая оболочка для данных *Lm-sensors*, позволяющая создавать собственную приборную панель.

SugarCRM: Как

ЧАСТЬ 3: В последней статье о SugarCRM **Марк Бэйн** расскажет, как добавить в это приложение собственные функции и освоить его по-настоящему.

Наш эксперт

Марк Бэйн
Свою первую программу Марк написал еще для ZX81. Он много лет работает с Unix и Oracle, а также читает лекции и пишет книги. Его вторая книга, «Руководство разработчика SugarCRM» («The SugarCRM Developer's Manual»), скоро выйдет из печати. (www.packtbub.com).

Если вы устанавливали и настраивали *SugarCRM* вместе с нами или делали это самостоятельно, у вас уже должна быть работающая версия программы с подходящей темой и терминологией, принятой у вас в организации, вместо стандартных «продаж и поддержки». Но удовлетворяет ли ее работа всем потребностям пользователей?

Вероятно, нет. В конце концов, *SugarCRM* предлагает универсальную базовую конфигурацию для всех типов организаций, и самое время подумать о том, как приспособить его для использования в вашей. Я покажу вам, как глубже проникнуть в таинство настройки и добавить в программу новые функции, необходимые для работы с клиентами. Мы добавим новый модуль для генерации отчетов и выпадающий список для хранения информации, не предусмотренной в *SugarCRM* по умолчанию.

Настройка выпадающего списка

Уж не знаю, как вы используете функцию Calls (Звонки), но обычно на вкладке Calendar (Календарь) нужно нажать на ссылку Schedule Call (Расписание телефонных звонков) и ввести данные о запланированных звонках.

Но что если часть этих контактов осуществляется не по телефону, а через программу обмена сообщениями (например, *Kopete*)? Как обозначить тип контакта в списке вызовов? Конечно, можно сделать соответствующую отметку в поле Описание (Description), но гораздо удобнее создать выпадающий список и задавать в нем тип контакта. Чтобы двигаться дальше, нам потребуются глубокие знания PHP? Отнюдь. Что тут думать: возьмем да просто сделаем это. В прошлый раз для переименования вкладок мы редактировали файл **custom/include/language/en_us.lang.php**. Добавим в этот же файл и новый выпадающий список с типами используемых контактов:

```
$app_list_strings['cmb_communication'] = array (
'TEL' => 'Telephone',
'MSN' => 'MSN',
);
```

Нетрудно видеть, что это массив, состоящий из:

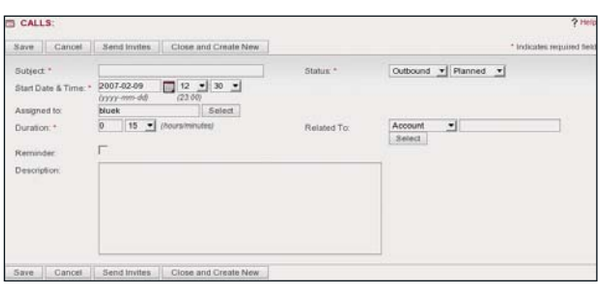
- Поля данных, определяющего хранимое в базе данных значение для каждого элемента.
- Текста, который будет отображаться на экране для этого элемента.

Выпадающий список готов.

Теперь определим его местоположение на странице: поработаем в *SugarCRM Studio*. Это компонент панели управления администратора, где можно заняться настройкой. Зайдите в *SugarCRM* как администратор (admin) и перейдите на экран Admin. Там расположена ссылка на *SugarCRM Studio*. Щелкнув на нее, выберите пункт меню Редактировать модуль (Edit A Module) (каждая вкладка, естественно, является отдельным модулем). Появится список модулей, которые можно редактировать; среди них есть и Calls. На странице этого модуля нажимаем кнопку Редактировать раскладку (Edit Layout), и перед вами предстанут раскладки на выбор: Detail View, Edit View, List View и Search Form. Выбираем Edit View.

Теперь преобразуем вкладку Edit View функции Calls. Справа располагается текущее содержимое вкладки, слева – элементы интерфейса, которые можно добавить на вкладку. Выпадающего списка, который мы добавили ранее, там пока нет. Чтобы он появился в левой части окна, нужно отвести ему поле в базе данных. Для этого достаточно нажать кнопку Add Custom Field (Добавить поле).

В появившемся окошке выберем тип нового поля. Первым делом надо указать тип данных – выберем Dropdown, поскольку нам нужен выпадающий список с перечнем типов контакта. Тут появится выпадающий список Drop Down List, содержимое которого мы определили ранее в файле **custom/include/language/en_us.lang.php**. Задайте полю базы данных понятное имя, например, db_communication. Также необходимо определить метку поля. Будьте внимательны: это не текст, кото-



› *SugarCRM* позволяет составить расписание звонков, но хватит ли на вкладке полей для всех необходимых данных?

Месяц назад Мы оформили *SugarCRM* в соответствии со стилем нашей марки.

добавить модули

рый будет отображаться рядом с выпадающим списком, а имя метки, так что укажите что-нибудь вроде lbl_communication (текстом мы займемся чуть позже). Поставим галочку Mass Update – это позволит нам вносить изменения не по одному, а все сразу.

Сразу после нажатия кнопки Сохранить (Save) в списке доступных компонентов в левой части экрана появится новое поле. Добавим в форму несколько пустых строк с помощью кнопки Добавить строки (Add Rows) и перетащим кнопку со списком в любое удобное место на форме. Затем еще раз нажмем кнопку Сохранить (Save) и получим возможность изменить текст таблицы (область для редактирования помечена «карандашиком»). Наконец, нажимаем кнопку Сохранить и опубликовать (Save And Publish). Готово. Теперь можно зайти на вкладку Календарь (Calendar tab), открыть список контактов и убедиться в том, что тип контакта теперь можно выбирать из списка. Полегчало.

Конечно, добавление новых элементов управления на вкладки – дело хорошее, но у вас могут оставаться задачи, совершенно не учтенные в *SugarCRM*. Пора задуматься о добавлении новых модулей. Как вы помните, каждый модуль *SugarCRM* предназначен для решения конкретной задачи – Управление счетами (Accounts), Перспективы (Opportunities), Проекты (Projects) и т.д. Соответственно, для решения конкретной задачи в нашей организации потребуется собственный модуль.

Прелести именования

Модуль – это всего лишь каталог на вашем web-сервере, содержащий PHP-скрипты. Поэтому первым делом нужно создать новый каталог. Но как же его назвать? Для начала посмотрим, какие модули уже установлены:

```
bainm@acamas:/www/SugarOS-Full-4.5.1$ ls modules
Accounts
ACL
ACLActions
ACLRoles
Activities
Administration
Audit
BeanDictionary.php
Bugs
Calendar
Calls
CampaignLog
Campaigns
...
```

Я привел лишь начало списка, но нетрудно заметить, что названия модулей *SugarCRM* либо используют терминологию из области продаж и сервиса (что вполне понятно), либо как-то связаны с выполняемой функцией (тоже понятно). Так бы и нам надо сделать, хотя и не совсем. Имя и вправду должно быть разумным, но не мешает заранее исключить накладки при обновлениях программы.

Например, в более ранней версии *SugarCRM 3.5.1* список модулей выглядел бы по-другому:

```
bainm@acamas:/www/SugarSuite-3.5.1$ ls modules
Accounts
Activities
Administration
Audit
BeanDictionary.php
Bugs
Calendar
Calls
Campaigns
Cases
...
```

И если бы мы добавили модуль '*CampaignLog*' в этой версии программы, а потом обновили бы ее до 4.5.1, то напоролись бы на его тезку. Следовательно, нужно выбрать название каталога таким образом, чтобы вероятность конфликта была минимальной. Вряд ли когда-нибудь появится отдельный модуль для журнала, даже такого известного, как Linux Format – вот вам и имя :

```
mkdir modules/LXF
```

Мы потратили кучу времени на такую простую задачу, как создание каталога. Однако правильный подход к этому делу избавит нас от неприятностей в будущем.

Наконец, мы разобрались с именем каталога и создали его. Оповестим *SugarCRM*, что в его полку модулей прибыло. Для этого добавим в конец файла **include/modules.php** следующую строку:

```
$moduleList[] = 'LXF';
```

Строку можно вставить в любое место файла, но я рекомендовал бы именно добавить ее в конец: потом проще будет ее найти.

Теперь мы должны сообщить *SugarCRM* еще кое-какую информацию о нашем модуле. Во-первых, нужен заголовок для соответствующей модулю вкладки. В прошлый раз мы уже занимались переименованием существующих вкладок, редактируя файл **custom/include/language/en_us.lang.php** – достаточно добавить в него новую строку для нашего модуля:

```
$app_list_strings['moduleList']['LXF'] = 'LXF';
```

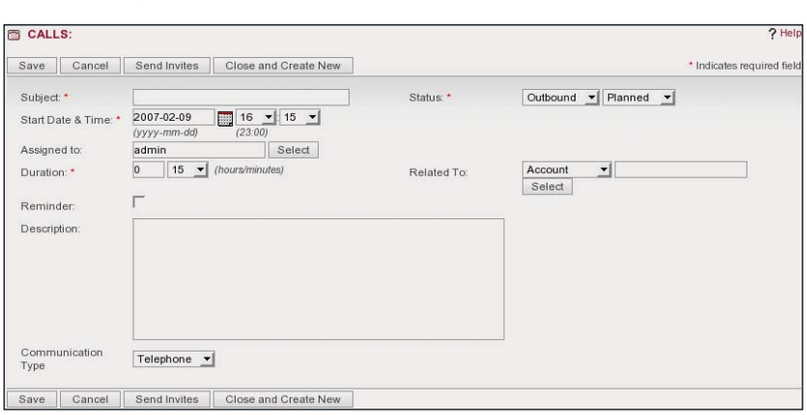
Наконец, нужно создать три файла. Их содержание сейчас безразлично – пусть будут хоть пустыми, лишь бы были (иначе вы не оберетесь ошибок). Чтобы модуль ожил и заработал, выполним команды

```
touch modules/LXF/Forms.php
touch modules/LXF/language/en_us.lang.php
touch modules/LXF/index.php
```

Попробуем открыть созданный модуль *SugarCRM* в web-браузере; если приложение уже открыто, просто обновите страничку. Должна появиться вкладка LXF. Выбрав ее, вы увидите… да ничего не увидите. Вкладка пуста. Оно и понятно: ведь мы еще не определили, что будет делать этот модуль. Этим сейчас и займемся.

Скорая помощь

Прежде чем писать собственный модуль, проверьте: может быть, такой уже есть на сайте www.sugarforge.org



Добавление нового поля («Тип контакта» в левом верхнем углу) займет всего пару минут, причем без всякого программирования.

» Задать модулю работу

Вы наверняка уже догадались, что дальше мы будем работать с файлом **modules/LXF/index.php**. Начнем с добавки в него чего-нибудь простенького:

```
<?php
function test_print () {
echo 'Hello Mum';
}
test_print();
?>
```

Не особо оригинально, но по крайней мере, обновив страничку в браузере, мы увидим, что запросто смогли отобразить необходимую информацию на вкладке.

Сделаем ее поинтереснее. Так как наш интерес – настройка *SugarCRM*, хорошо бы модуль просмотрел базу данных приложения. Начнем с функции, которая выводит список таблиц базы данных:

```
function show_tables () {
global $sugar_config;
```

Выражение global обеспечивает доступ к глобальной переменной $sugar_config внутри функции. А что такое $sugar_config? Это массив, содержащий все настройки *SugarCRM*, которые определены в файле config.php в каталоге программы. Нам нужны настройки для соединения с базой данных, их можно, получить, например, таким образом:

```
$username=$sugar_config['dbconfig']['db_user_name'];
$password=$sugar_config['dbconfig']['db_password'];
$database=$sugar_config['dbconfig']['db_name'];
```

Затем мы можем установить соединение с базой данных:

```
$db = mysql_connect(localhost,$username,$password);
mysql_select_db($database) or die( "Unable to select database");
```

После установки соединения можно выполнить запрос к базе данных, в данном случае, получение списка таблиц:

```
$query="show tables";
$result=mysql_query($query);
```

Добавим цикл для вывода на экран названий таблиц:

```
echo "<table width=100% cellspacing=0 cellpadding=0>";
$r=0;
while ($r < mysql_numrows($result)) {
   $tablename = mysql_result($result,$r);
   echo "<tr><td>$tablename</td></tr>";
   $r++;
}
echo "</table>";
```

Наконец, хорошим тоном будет закрыть соединение с базой данных:

```
mysql_close($db);
}
```

Функция создана – можно ее вызывать:

```
show_tables();
```

Результатом ее работы станет появление на экране списка из 92 таблиц. Занятно, но хотелось бы по щелчку на ссылке получать и описание конкретной таблицы. Для этого изменим определение функции на следующее:

```
function show_tables ($table, $table_action) {
```

Теперь можно передать в функцию параметр $table, определяющий вид запроса. Заменим строку запроса

```
$query="show tables";
```

на следующие:

```
if ($table && $table_action=="desc") {
$query="desc $table";
} else {
$query="show tables";
}
```

То, что выводится на экран, должно зависеть от значения $table. Поэтому код

```
while ($r < mysql_numrows($result)) {
  $table=mysql_result($result,$r);
  echo "<tr><td>$tablename</td></tr>";
  $r++;
}
```

превратится в

```
while ($r < mysql_numrows($result)) {
   if ($table && $table_action=="desc") {
      $field = mysql_result($result,$r,0);
      $type = mysql_result($result,$r,1);
      $null = mysql_result($result,$r,2);
      $key = mysql_result($result,$r,3);
      $default = mysql_result($result,$r,4);
      $extra = mysql_result($result,$r,5);
      echo "<tr><td>$field</td><td>$type</td><td>$null</td><td>
$key</td><td>$default</td><td>$extra</td>";
      echo "</tr>";
} else {
      $tablename = mysql_result($result,$r);
      echo "<td><a href=index.php?module=LXF&action=index&table=$ta
ble
name&table_action=desc>
      View data definition for $tablename</a></td>";
      echo "</tr>";
   }
   $r++;
}
```

Легко видеть, что вывод может быть разным. Если переменная $table не определена, на экране отображается полный список таблиц базы данных. В противном случае, отображается описание конкретной таблицы.

Приведенный код также дает представление о том, как модулям *SugarCRM* передается информация. Файл **index.php**, находящийся в каталоге *SugarCRM*, формирует интерфейс пользователя *SugarCRM*. Этот файл принимает два главных параметра:

- **module** Имя модуля.
- **action** Имя PHP-файла, вызываемого из каталога модуля.

Вы можете передать ему и другую информацию. Для получения списка таблиц я использовал следующий URL ('acamas' – название моего веб-сервера)

```
http://acamas/SugarOS-Full-4.5.1/index.php?module=LXF&action=index
```

а для получения описания таблицы, например, cases, следующий URL:

```
http://acamas/SugarOS-Full-4.5.1/index.php?module=LXF&action=index&
table=cases&table_action=desc
```

Наконец, подправим наш код, чтобы он был в курсе параметров, переданных модулю:

```
show_tables($_REQUEST['table'], $_REQUEST['table_action']);
```

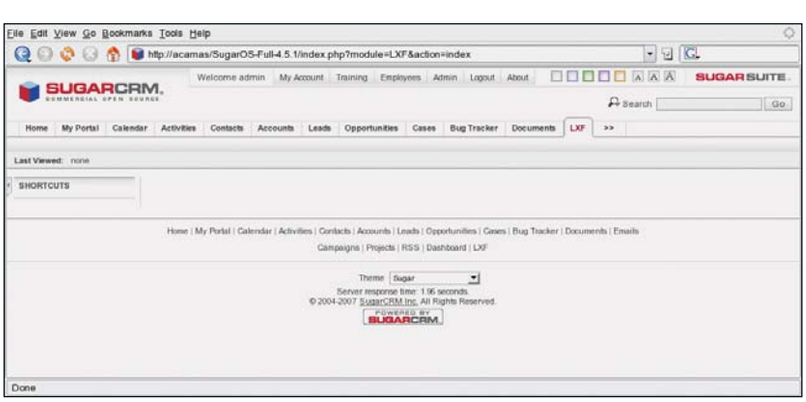
Войдя во вкус, вы захотите увидеть еще и содержимое таблиц; значит, код опять придется немного изменить. Во-первых, добавим ссылку, которая передаст необходимую информацию модулю. Вот она:

```
echo "<td><a href=index.php?module=LXF&action=index&table=$table
name&table_action=contents>Table contents</a></td>";
```

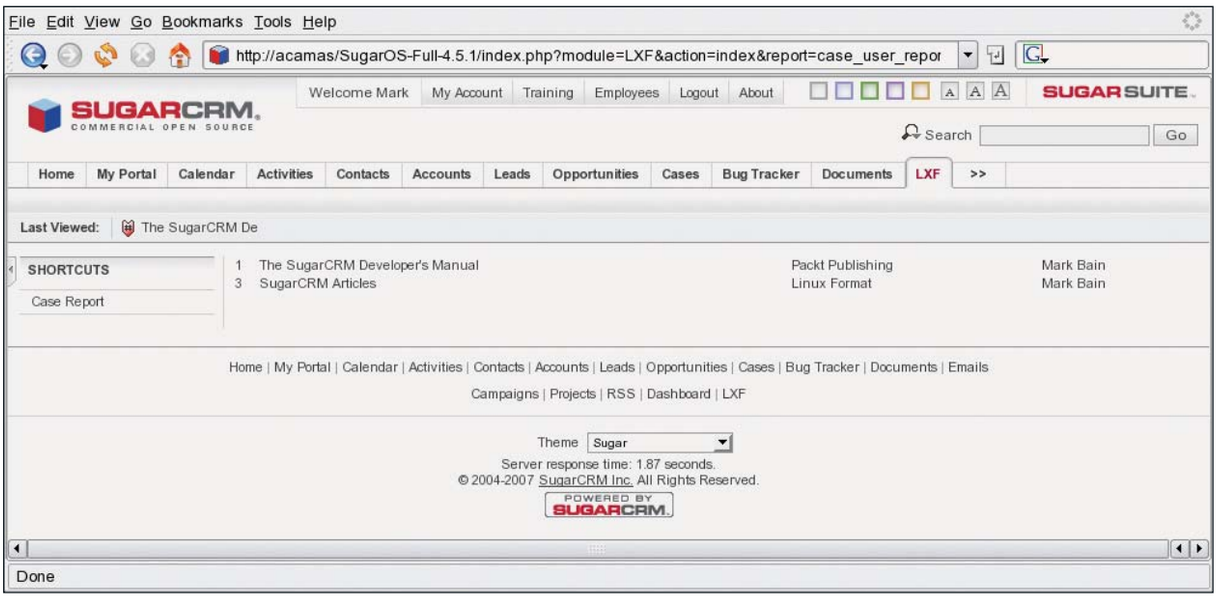
Затем добавим немного кода SQL для получения содержимого таблиц:

Скорая помощь

По умолчанию выход пользователя из приложения SugarCRM происходит через 1 440 секунд (24 минуты) простоя. Если вам это неудобно, измените переменную session.gc_maxlifetime в файле **/etc/php4/apache2/php.ini**. Не забудьте перезапустить Apache.



› Новый модуль 'LXF' в ожидании дальнейших действий.



› Цель нашего урока: собственный полностью функциональный модуль

```
if ($table && $table_action==”desc”) {
  $query=”desc $table”;
} elseif ($table && $table_action==”contents”) {
  $query=”select * from $table”;
} else {
  $query=”show tables”;
}
```

и обработаем полученную информацию:

```
while ($r < mysql_numrows($result)) {
```

Здесь менять нечего, а вот дальше уже интереснее.

```
} elseif ($table && $table_action==”contents”) {
  echo “<tr>”;
  $c=0;
  while ($c < mysql_num_fields($result)) {
    $field=mysql_result($result,$r,$c);
    echo “<td>$field</td>”;
    $c++;
  }
  echo “<tr>”;
} else {
```

Вы заметили, что получение данных таблицы происходит не совсем так, как ее описания. Здесь только 6 полей (они пронумерованы от 0 до 5), а SQL-запрос возвращает произвольное их количество, и это нужно учесть.

Вы недоумеваете, зачем нужно было создавать вкладку для отображения содержимого таблиц? (Вообще-то пользователям лучше даже не подозревать о базе данных). Дело в том, что она поможет вам создавать отчеты для ваших пользователей.

Генератор отчетов

Чтобы пояснить то, что я имею в виду, давайте создадим отчет, который будет содержать информацию по всем открытым сделкам (cases). Достаточно будет изменить текст SQL-запроса примерно таким образом:

```
} elseif ($report==”case_user_report”) {
  $query=”select c.case_number, c.name, a.name, concat(u.first_
name, concat(‘ ‘,u.last_name))
from cases c, users u, accounts a
where u.id = c.assigned_user_id
and a.id = c.account_id
and c.status <> ‘Closed’”;
```

Теперь мы должны обработать результат. У нас уже есть подходящий код для обработки запроса выборки, остается только задействовать его и для отчетов:

```
} elseif ($report || ($table && $table_action==”contents”)) {
```

Доделайте работу сами (хотя весь код этого примера есть на диске). А когда закончите, сможете просмотреть отчет, используя следующий URL (не забудьте передать в функцию параметр ‘report’, а внутри функции прочитать его, используя $_REQUEST[‘report’]):

```
http://acamas/SugarOS-Full-4.5.1/index.php?module=LXF&action
=index&report=case_user_report
```

Все работает, хотя URL довольно неуклюжий. Гораздо удобнее использовать специальную ссылку.

Разбираемся со ссылками

С левой стороны вкладки нашего модуля расположена панель ссылок. Пока на ней ничего нет, кроме заголовка Shortcuts (Ссылки). Постараемся, чтобы от нее была польза.

Для этого создадим файл **Menu.php** в каталоге модуля (то есть **modules/LXF/Menu.php**). Этот массив массивов. Каждый его подмассив содержит собственно ссылку и текст, который будет отображаться на экране. Например:

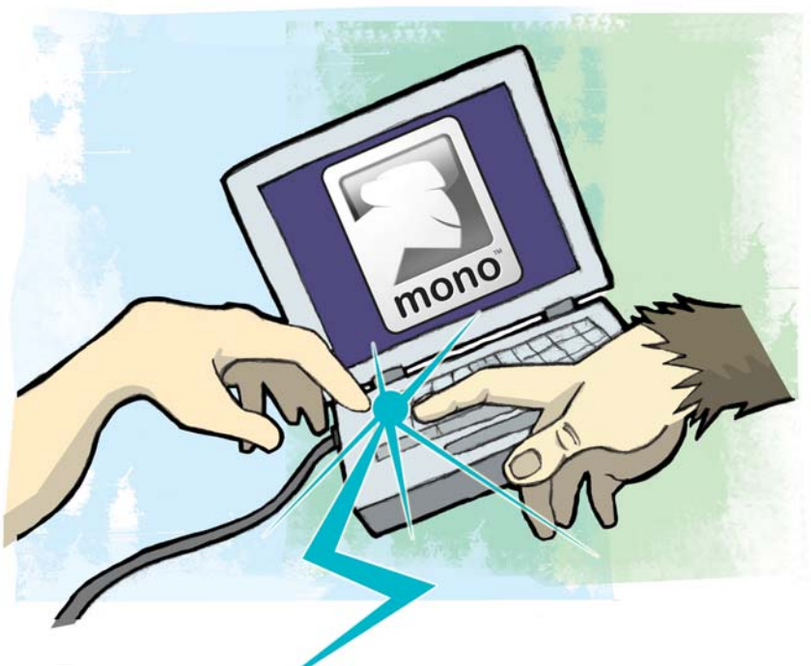
```
<?php
$module_menu = Array(Array(“index.php?module=LXF&action=index&re
port=case_ user_report”, ‘Case Report’),);
?>
```

Сохраните этот файл и обновите страничку в браузере. Теперь можно обращаться к генератору отчетов по ссылке.

Итак, мы полностью настроили *SugarCRM* в соответствии с потребностями пользователей. Или не полностью? Но все равно, у нас есть все необходимое для дальнейшего расширения пользовательской функциональности. Счастливых настроек! LXF

Mono: Объекты и

Объектно-ориентированное программирование кое-кого пугает больше, чем школьников прививки, но **Пол Хадсон** собирается обойтись без боли.



Наш эксперт

Пол Хадсон полагает, что Mono – лучшая вещь со времен мультфильма Pinky and the Brain, и сейчас поддерживает два проекта на основе Mono на SourceForge

В теории, теория и практика совпадают, но на практике так бывает редко. Вот почему в наших уроках обучение строится на процессе выполнения: если вы читали этот учебник с самого начала, то уже сделали пять полноценных приложений, решающих реальные задачи. На данном уроке я хочу отойти от принятой формулы и обсудить довольно сложную теорию программирования: объектно-ориентированное программирование (ООП) и обобщенные типы.

Это не штуки, из которых можно делать свои приложения, а просто методы, которые пригодятся вам при программировании на Mono. Коль скоро вы это поняли, я покажу вам, как реализовать карточную игру с помощью названных двух методов.

Классификация объектов

ООП позволяет определять предметы в вашем программном коде и даже придавать им желаемое поведение. Вы уже использовали ООП, только не догадывались об этом. Создайте новый проект с именем Geno (еще один редкий персонаж Nintendo – уж простите!), и вы увидите, что MonoDevelop напишет код по умолчанию:

```
class MainClass
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```

Здесь применяется ООП, и программа отлично работает, даже если вы не понимаете, что это значит (да и знать не хотите). Но теперь знайте: класс – это определение предмета, а объект – это экземпляр предмета. Ясно как ночь? Так вот: на вопросы «Какого цвета машина?», «Какая длина у машины?» или «Сколько у машины передач?» ответ будет «это зависит от»: что такое машина, представляют все, но каждая машина индивидуальна.

В терминах ООП, «машина» является классом. Но «машину вообще» увидеть нельзя: это абстрактное понятие. На самом деле мы видим «Форды», «Хонды» и так далее, то есть физические реализации класса «машина». Итак: машина, находящаяся на шоссе, это объект класса «машина». У нее есть цвет, длина, и вы знаете, сколько у нее передач, но это просто переменные свойства класса «машина». Другие машины, даже той марки, что и ваша, могут сильно отличаться; но все они машины.

Если для вас это все еще пустой звук, подождите: вы все поймете из кода! А пока *MonoDevelop* определил для нас класс MainClass, содержащий метод public static void Main(), который мы все время используем. Два странных слова – public и static – относятся к ООП.

Объект Geno

Измените MainClass на Geno, имя нашего проекта. Теперь замените строку Console.WriteLine() на следующую:

```
Geno game = new Geno();
```

Этот код создает объект класса Geno и присваивает его переменной game. Что представляет собой класс Geno? В данный момент он содержит только Main() и больше ничего – это просто пустая переменная. Но она рождает интересный вопрос: строка находится внутри метода Main(), который находится внутри класса Geno. Как может Geno создать сам себя? Или – основной вопрос философии: что появилось раньше, класс Geno или метод Main()?

Тут на помощь приходит слово 'static' – статический. Метод Main(), если вы помните, помечен как public static void, и на то есть причина: статические методы могут вызываться без экземпляра класса. Фактически они привязывают метод к классу, просто в организационных целях. Например, если в нашем классе был метод СменитьПередачу(), его применение не имело бы смысла без конкретного экземпляра машины, так как в противном случае, какую передачу надо менять? А как насчет вычисления тормозного пути машины, мчащейся со скоростью 100 км/ч? К машинам это имеет отношение, но ясно, что для этого не требуется реальный объект машины.

Таким образом, статический метод Main() может вызываться без существования класса Geno, и мы используем его для создания объекта Geno так, чтобы можно было играть в карточную игру. Объект Geno будет контролировать все аспекты игры, поэтому других объектов нам не понадобится. Но, ради интереса, мы добавим еще два класса: один будет отвечать за игроков, другой за карты. Логика отдается на откуп объекту Geno, так что классы игрока и карт предназначены просто для хранения данных.

Есть еще кое-что, что вам надо знать, прежде чем писать код. Иногда нужно, чтобы переменная принимала значение только из определенного набора. Например, набор данных для дней недели – воскре-

Месяц назад Мы применили GTK# для чтения RSS-лент.

обобщенные типы

сенье, понедельник, вторник и т.д. Для карточной игры требуется, чтобы каждая карта была определенной масти: червы, бубны, трефы или пики. C# позволяет определить масти карт как перечисление:

```
public enum Suits { Hearts, Diamonds, Clubs, Spades };
```

Описываем игру

Запрограммируем детскую игру: она, возможно, знакома вам как «Дама червей». Из карточной колоды извлекается одна дама (бубен), остается 51 карта. Карты сдаются всем игрокам, втемную. Игроки смотрят на свои карты и сбрасывают пары карт одинакового достоинства: например, если у игрока есть две десятки, то он кладет эти две десятки на стол. Дама червей не может быть использована как парная карта; игрок, которому она досталась, должен ее сохранить.

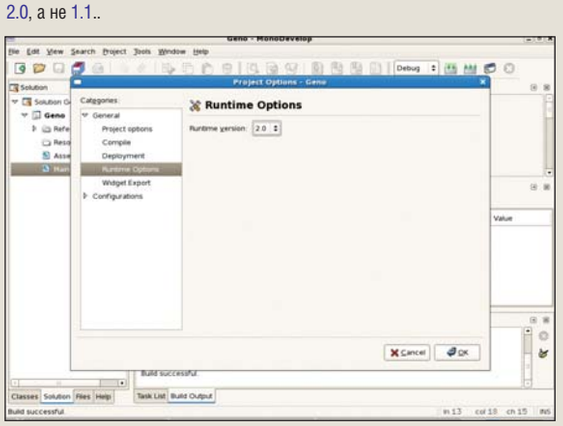
После того, как все игроки выкинули свои пары, первый игрок поворачивается к игроку справа и втемную забирает у него произвольную карту. Если в результате у игрока образовались парные карты, то он может их сбросить. Игра продолжается, и второй игрок поворачивается к игроку справа и вынимает карту – и так далее. В конечном счете, каждая карта должна найти себе пару, за исключением дамы червей, а игрок, у которого она на руках, проигрывает. [по-русски эта игра называется «Акулина» или «Акулька», но непарная дама – пиковая, – прим.ред.]

Нам надо предусмотреть следующие функции:

- Play() Эта функция начинает игру, после проведения необходимых настроек.
- ShuffleCards() Перетасовать колоду (рандомизировать порядок карт).
- RemovePlayerPairs() Поиск и удаление подходящей пары карты у игрока.
- PrintResult() Печать результатов игры (у кого осталась червонная дама).

Важное замечание

При работе с MonoDevelop проверьте, что он использует среду .NET 2.0, в противном случае столкнетесь с проблемами. Вам надо изменить настройки для каждого проекта – выберите Project > Options, затем выберите категорию Runtime Options и установите среду выполнения на 2.0, а не 1.1..



Наш проект требует совместимости со средой .NET 2.0. Версия Mono из Fedora подойдет.

Также потребуется определить классы Player и Card, которые будут содержать информацию. Вот скелет будущего кода [для большей ясности: Suit – масть, Card – карта, Hearts – черви, Diamonds – бубны, Clubs – трефы, они же кресты, Spades – пики, Player – игрок, англ.]:

```
using System;
namespace Geno {
enum Suits { Hearts, Diamonds, Clubs, Spades };
class Card {
    public int Val;
    public Suits Suit;
}
class Player {
    public int Score;
}
class Geno {
    static void Main(string[] args) {
        Geno game = new Geno();
        game.Play();
    }
    void Play() { }
    void ShuffleCards() { }
    void RemovePlayerPairs(int player) { }
    void PrintResults() { }
}
}
```

Ну да, знаю, здесь куча пустых методов, но они проясняют структуру программы. Заметили, что мне пришлось объявить все переменные внутри классов Player и Card как public? Это потому, что переменные внутри объекта доступны только внутри самого объекта, чтобы внешние части кода их не затрагивали. Но так как у нас довольно простая программа, класс Geno будет делать большую часть работы и использовать классы Player и Card просто для хранения значений. То, что эти переменные стали public, значит, что класс Geno может их читать и записывать [в серьезных программах так делать не рекомендуется – вместо этого следует определить методы, обеспечивающие доступ извне к закрытым переменным, – прим.ред.].

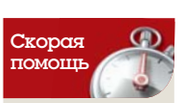
Введение в обобщенные типы

У нас есть класс Card, то есть мы можем определить значение карты (от 1 до 13) и ее масть. Но мы еще не создали сам объект «карты» и не знаем, где их хранить. Здесь приходят на помощь массивы: они позволяют хранить множество объектов в одной переменной. У Mono есть огромное количество типов массивов, но долгое время наиболее используемым был ArrayList. Он позволяет хранить в массиве любой тип объекта и обращаться к нему просто по индексу. Но здесь есть проблема: у каждой переменной в C# есть тип, будь то integer, string, Suit или любой другой. Так как ArrayList может содержать переменную любого типа, то вам всегда придется сообщать Mono, какой тип используется. Например:

```
int i = 1;
ArrayList numbers = new ArrayList();
numbers.Add(i);
int j = numbers[0];
```

Здесь будет ошибка – Mono не сможет преобразовать numbers[0] в integer, даже если мы знаем, что оно уже типа integer. Вместо этого надо написать:

```
int j = (int)numbers[0];
```



Если вы не хотите, чтобы переменная была недоступна вне класса, удалите слово public. Или, если хотите быть точным, напишите вместо него слово private.

» Префикс (int) значит «обращаться с numbers[0] как с целым числом», и наш код будет компилироваться правильно. Новые версии C#, включая поставляемую с Fedora Core 6, поддерживает новый способ программирования, известный как обобщенные типы (generics). И если вы когда-либо раньше использовали C++, то знакомы с термином «шаблон», а это почти тоже самое – только на вид гораздо легче!

Обобщенные типы – это произвольные массивы, которые принимают только один тип переменных. Вам уже не надо приписывать (int), чтобы вытаскивать целые числа из обобщенного списка – туда так и так можно помещать только целые числа. В Geno обобщенные типы будут использоваться для двух вещей: хранения карт и хранения игроков. У каждого игрока будет свой собственный список карт, так как карты из колоды раздаются именно игрокам.

Добавьте такие две строчки сразу под строкой class Geno:

```
List<Card> Cards = new List<Card>();
List<Player> Players = new List<Player>();
```

Этот синтаксис может затуманить мозги, но по-простому он гласит «Хочу, чтобы один список содержал переменные типа Card, а второй список содержал переменные типа Players». Вы также должны добавить строку до переменной Score в классе Player:

```
public List<Card> Cards = new List<Card>();
```

Скорая помощь

Если в вашем проекте много классов, может оказаться, что они начнут ограничивать имена используемых переменных. Одно из решений – поместить один класс в другой, то есть сделать вложенный класс: если класс A находится внутри B, только класс B может обращаться к классу A.

Устанавливаем игру

Оба наших списка Players и Cards пусты, поэтому первым заданием будет поместить 51 карту в колоду (помните, что мы убрали бубновую даму) и разместить несколько игроков. Самым простым способом вставить карты будет перебрать в цикле все масти (Suits), и для каждой масти посчитать от одного до 13 так, чтобы получить все от тузов до королей [туз считается единицей]. Добавив все карты и всех игроков, завершаем установку, вызывая метод ShuffleCards() для перетасовки карт.

```
foreach (Suits suit in Enum.GetValues(typeof(Suits))) {
    for (int i = 1; i < 14; ++i) {
        Card card = new Card();
        card.Val = i;
        card.Suit = suit;
        if (card.Val == 12 && card.Suit == Suits.Diamonds) continue;
       Cards.Add(card);
    }
}
for (int i = 0; i < 4; ++i) {
    Player player = new Player();
    Players.Add(player);
}
ShuffleCards();
```

Вы видите, что надо вызвать просто Cards.Add(card), чтобы вставить карту в список из Cards; все очень просто. Метод ShuffleCards() пока ничего не делает, потому что он пустой. Нам необходимо пройтись по всему массиву Cards, вытащить отдельные карты и поместить их обратно в произвольную позицию. Тут не обойтись без генератора случайных чисел – вставьте эту строку перед static void Main:

```
Random Rand = new Random();
            Теперь полная реализация ShuffleCards():
void ShuffleCards() {
   for (int i = 0; i < Cards.Count; ++i) {
       Card tmp = Cards[i];
       Cards.RemoveAt(i);
       Cards.Insert(Rand.Next(0, Cards.Count), tmp);
   }
}
```

Здесь показано несколько новых возможностей списков: у них есть свойство Count, которое возвращает число содержащихся в них элементов; они содержат метод RemoveAt(), который удаляет элемент в указанной позиции; и у них есть метод Insert(), который вставляет элемент в указанную позицию. Номер позиции определяется переменной Rand, которая может генерировать число между 0 и Crads.Count (количество карт в колоде).

Раздаем карты

Наша колода заполнена и стасована. Осталось сдать ее игрокам. Чтобы это сделать, будем давать им карты, пока колода не кончится. Это значит, что нам надо начать с игрока 0 (в C# списки начинаются с 0), сдать карту, перейти к игроку 1, сдать карту и так далее, пока не закончатся игроки; потом перейти снова к игроку 0. В коде этот алгоритм будет выглядеть вот так:

```
int playernum = 0;
while (Cards.Count > 0) {
   Players[playernum].Cards.Add(Cards[0]);
   Cards.RemoveAt(0);
   ++playernum;
   if (playernum == Players.Count) playernum = 0;
}
// удалять начальные пары
for (int i = 0; i < Players.Count; ++i) {
   RemovePlayerPairs(i);
}
```

В последней части (от комментариев и ниже) уже начинается логика игры: каждый игрок удаляет пары карт, которые оказались у него в начале игры.

Пишем логику

Метод RemovePlayerPairs() принимает на вход номер игрока и удаляет у него парные карты. Чтобы облегчить понимание, я разделил функцию на две, вот так:

```
void RemovePlayerPairs(int player) {
   while (TryPairRemove(player)) {
    ++Players[player].Score;
   }
}
```

Итак, номер игрока поступает в функцию и передается методу TryPairRemove(). Если этот метод вернул true, значит, была найдена пара; затем он вызывается снова. В конце концов будут найдены все пары, и цикл завершит свою работу.

Метод TryPairRemove() немного сложноват, так как ему надо у каждого игрока взять карту, перебрать остальные его карты на предмет совпадения, и если в паре ни одна из карт не является дамой червей – удалить пару и вернуть true.

```
bool TryPairRemove(int player) {
    Card card1;
    Card card2;
    Player thisplayer = Players[player];
    for (int i = 0; i < thisplayer.Cards.Count; ++i) {
      card1 = thisplayer.Cards[i];
      if (card1.Suit == Suits.Hearts && card1.Val == 12) continue;
      for (int j = i + 1; j < thisplayer.Cards.Count; ++j) {
          card2 = thisplayer.Cards[j];
          if (card2.Suit == Suits.Hearts && card2.Val == 12) continue;
          if (card1.Val == card2.Val) {
                thisplayer.Cards.RemoveAt(j);
                thisplayer.Cards.RemoveAt(i);
                Console.WriteLine("Player " + (player + 1) + " plays " +
                CardName(card1) + " and " + CardName(card2));
                return true;
          }
      }
   }
   return false;
}
```

Некоторые комментарии по коду:

1. j начинается с i+1: среди проверенных карт совпадений нет.
2. Проверка совпадения с дамой червей делается и для card1, и для card2.
3. Если найдено совпадение, сначала удаляется j, затем i. Удаление элемента из списка заставляет сдвигаться все элементы массива,

Попробуйте другие типы

Списки – это несортированные массивы, обращение к которым осуществляется по номеру позиции, но можно также использовать словари, с обращением через определенный вами тип. Например, Dictionary<string,string> хранит строки как ключи массива и как его значения. Если ключ foo имеет значение bar, вы можете найти его так: MyDictionary[“foo”]. Можете использовать любой тип данных – даже объекты или другие обобщенные типы, если хотите.

чтобы закрыть пробел, поэтому первым надо удалять элемент с более высоким индексом.

4 Метод CardName() будет описан далее.

Да, это солидный кусок кода, ведь мы должны дважды перебрать карты игрока. Метод CardName() очень прост и выдает названия карт, я не буду приводить его здесь – можете обратиться к исходному коду на диске.

Последний рывок

Оставшийся код обрабатывает основной игровой цикл: игроки выбирают карты и пытаются найти пары. Самой сложной частью является выбор игрока, у которого надо вытащить карту: код должен начать искать игрока «справа» от нас (то есть его номер должен быть больше нашего), но если он никого не находит, то начинает смотреть с начала списка. Если он вернулся обратно, значит, игра закончилась. Если найден подходящий игрок, у него выбирается карта, возможно, сбрасываются новые пары, и игра продолжается.

Этот код должен следовать до конца метода Play(). Он длинный, но не такой сложный, если разбить его на части вот так:

```
bool finished = false;
while (!finished) {
    int playersleft = Players.Count;
    for (int i = 0; i < Players.Count; ++i) {
        Player player = Players[i];
        if (player.Cards.Count == 0) {
            --playersleft;
            continue;
}
```

Здесь отслеживается число игроков, оставшихся в игре. Если это значение равно 1, то надо выходить; займемся этим позже. Сейчас цикл просто обрабатывает каждого игрока.

```
int playertouse = -1;
for (int j = i + 1; j < Players.Count; ++j) {
    if (Players[j].Cards.Count > 0) {
        playertouse = j;
        break;
    }
}
if (playertouse == -1) {
    for (int j = 0; j < Players.Count; ++j) {
    if (Players[j].Cards.Count > 0) {
        playertouse = j;
        break;
    }
  }
}
```

Этот блок ищет игрока, у которого нужно вытянуть карту: сперва над текущей позицией, потом с начала. Игроки, у которых нет карт, автоматически пропускаются.

```
if (playertouse == i) {
  finished = true;
  break;
} else {
   int cardtochoose = Rand.Next(0, Players[playertouse].Cards.
Count);
   Players[i].Cards.Add(Players[playertouse].
Cards[cardtochoose]);
   Players[playertouse].Cards.RemoveAt(cardtochoose);
   RemovePlayerPairs(i);
  }
}
```

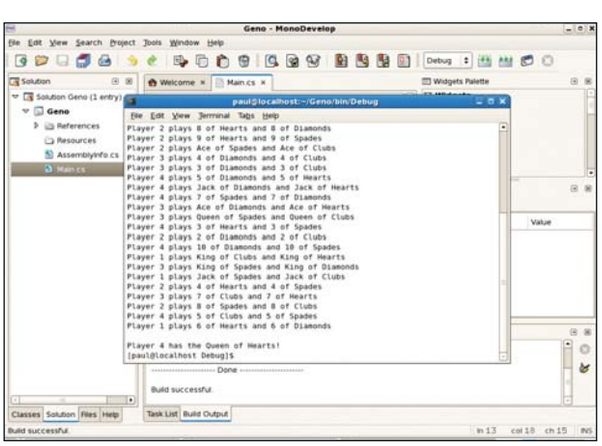
Теперь начинается самое интересное: если мы добрались сами до себя, значит, нет игроков, у которых можно взять карту. В противном случае, берем произвольную карту и вызываем метод RemovePlayerPairs(), для сброса подходящей пары.

```
if (playersleft == 1) finished = true;
}
PrintResults();
```

Наконец, если игроков не осталось, завершаем цикл. Игра закончена, и PrintResult() выдает сообщение:

```
void PrintResults() {
    Console.WriteLine(“”);
    for (int i = 0; i < Players.Count; ++i) {
    if (Players[i].Cards.Count > 0) {
      Console.WriteLine(“У игрока “ + (i + 1) + “ осталась дама
червей!”);
      break;
    }
  }
}
```

Если вы запустите программу, то увидите четырех компьютерных игроков, играющих около секунды. Теперь у вас есть понимание о классах и объектах, и вы можете извлечь преимущества обобщенных типов для хранения ваших объектов. Вы также узнали, как сделать несложную карточную игру – полпути к разработке покера или любой другой игры. LXF



› Конечный результат не блестящ: консольная программа выглядит просто, но требует много работы (и обучения).

Скорая помощь

Есть соблазн использовать целые числа, а не перечисления для списков. Например, воскресенье может быть 0, понедельник – 1, и так далее. Хотя для небольших списков это работает, в случае больших списков легко запутаться. Перечисления работают ничуть не медленнее целых, так как автоматически преобразуются в целые числа.

» **Через месяц** Объектное ориентирование, часть 2: наследование.

Ant: Упростим

Если вы используете Java, то без Ant вам не обойтись. **Скотт Дуглас** покажет, как упростить дистрибуцию и облегчить процесс Java-разработки.

Наш эксперт

Скотт Дуглас
Скотт – разработчик программного обеспечения, который специализируется на Java. Он использует Linux уже 7 лет и полагает, что Vim – это круто.

Если вы используете Java не только для запуска *Azureus*, то, скорее всего, встречались с *Apache Ant*. Возможно, вы использовали его для компиляции скачанного Java-пакета или писали с его помощью файл сборки для ваших собственных проектов. *Ant* [англ. муравей, – прим. пер.]. стал инструментом де-факто для сборки всего на Java. Он берет на себя все труды по компиляции Java-проектов, и при правильном использовании управляет путями к классам и библиотеками.

Однако *Ant* умеет не только компилировать: он включает множество встроенных инструкций (известных как задачи), позволяющих легко управлять вашим Java-проектом. Одной из примечательных черт *Ant* является его способ упростить процесс сборки, разбивая каждую стадию на составные подзадачи, которые потом группируются для создания файла сборки проекта.

На этом уроке мы рассмотрим и выполним некоторые задачи, но сначала нам надо удостовериться, что ваш простой файл проекта правильно структурирован: легче будет добавлять новые задачи.

Мой первый файл проекта

```
<project name="id3" basedir=".">
<description>
Build file for id3 project
</description>
<!-- Properties -->
<property name="src" value="/home/art/projects/id3/src"/>
<property name="libs" value="/home/art/projects/id3/libs"/>
<!-- Classpath -->
<path id="base.path">
<pathelement path="${classpath}"/>
<fileset dir="${libs}">
<include name="**/*.jar"/>
</fileset>
</path>
<!-- Compile -->
<target name="compile" description="compile the source code">
<echo>Compiling source</echo>
<javac classpathref="base.path"
srcdir="${src}">
<include name="org/**"/>
</javac>
</target>
</project>
```

Это довольно стандартный файл сборки *Ant*. Он поможет нам с компиляцией проекта и с расположением классов и библиотек, но мало с чем еще. И такая работа, конечно, пригодится, но почему бы файл не улучшить? Для начала неплохо создать свойства вверху файла и обращаться к ним по имени из задач (например, так: ${имяСвойства}). Однако это не очень переносимый вариант: что если мы хотим собрать наш проект на другой системе или даже просто в другом каталоге?

Переносимые свойства

Вот что нам надо сделать: сохранить все свойства, относящиеся к путям, в отдельном файле и просто указать *Ant*, чтобы он читал их из этого файла. Первый шаг к достижению этого – замена двух строк, находящихся в разделе Properties, на одну следующую:

```
<!-- Properties -->
<property file="local.properties"/>
```

Для хранения наших свойств мы создали файл с именем **local.properties** в том же каталоге, что и наш файл сборки, и поместили в него все свойства, которые могут измениться. Это просто текстовый файл, хранящий пары вида Свойство=Значение; строки, начинающиеся с #, *Ant* игнорирует.

```
# Локальный файл свойств
# Содержит каталоги для сборки
src=/home/art/projects/id3/src
build=/home/art/projects/id3/build
libs=/home/art/projects/id3/libs
dist=/home/art/projects/id3/dist
etc=/home/art/projects/id3/etc
mainclass=org.sturgeon.Id3Renamer
jarfile=id3.jar
debug=true
fork=true
source=1.5
```

Поскольку наша задача «компиляции» уже использовала свойства, записанные в начале исходного файла, то для обращения к ним никаких изменений проводить не требуется. Однако, так же, как мы уста-

Месяц назад Мы помогли сообществу, собрав Deb-файлы для Ubuntu.

Java-проекты

новили локальные пути, можно добавить свойства для других вещей. Здесь, fork указывает *Ant*, использовать ли компилятор JDK внешним образом, source задает требуемый уровень исходного кода, а debug указывает *Ant*, следует ли включать отладочную информацию в файлы классов.

Использование внешнего файла свойств означает, что мы можем использовать тот же файл сборки на любых платформах без изменений – только для каждой установки надо предоставить файл **local.properties**. Вот задача компиляции с включенными новыми флагами свойств.

```
<!-- Compile -->
<target name="compile" description="compile the source code">
<echo>Compiling source</echo>
<javac srcdir="${src}"
destdir="${build}"
classpathref="base.path"
fork="${fork}"
debug="${debug}"
source="${source}"/>
<include name="org/**"/>
</javac>
</target>
```

Теперь мы можем добавить задачу для тестирования проекта – почти так же, как мы тестировали проект компиляции. Единственная существенная разница – надо будет изменить classpath так, чтобы он включал каталог сборки. Это не вопрос: мы уже определили базовый путь для проекта с помощью атрибута path. К нему можно обращаться внутри атрибута classpath, надо только добавить строчку каталога сборки. Стоит также упомянуть здесь использование fork: оно сообщает *Ant*, исполнять ли код в другой JVM (а не в той, в которой работает сам *Ant*). Если вы собираетесь посылать аргументы командной строки в JVM, установите его в Yes.

```
<!-- Test -->
<target name="test" description="test run the project">
<echo>Running project</echo>
<java classname="${mainclass}"
fork="${fork}"
dir="${build}">
<classpath>
<pathelement location="${build}"/>
<path refid="base.path"/>
</classpath>
</java>
</target>
```

Муравей на хозяйстве

Прежде чем упаковать наш код в JAR-файл, добавим-ка пару задач для упрощения структуры проекта. К сожалению, *Ant* не станет пылесосить за вас квартиру (зато если у вас есть *Roomba* [*Roomba* – робот-пылесос, см. на сайте **http://en.wikipedia.org/wiki/Roomba**, – прим. пер.], вы можете удумать умные задачи для его управления), но зато создаст нам структуру каталогов. Для начала припасем задачу Init – с ней более или менее ясно: все, что она делает – создает каталог **build** (для размещения файлов классов) и dist (где разместится JAR-файл). Далее предусмотрим задачу Clean, чтобы удалить ранее созданные каталоги. Задача Clean полезна для сборки во время различных стадий проекта, так как позволяет «начать с нуля» в любой момент. Добавьте эти две задачи в файл сборки, и мы перейдем к задаче Jar.

```
<!-- Init -->
<target name="init" description="initialise directories">
<echo>Initialising directories</echo>
<mkdir dir="${build}"/>
<mkdir dir="${dist}"/>
</target>
<!-- Clean -->
<target name="clean" description="remove directories">
<delete dir="${build}"/>
<delete dir="${dist}"/>
</target>
```

Основная задача Jar очевидна: здесь указываются свойства для каталога вывода, а также файлы, которые надо включить. Однако мы введем в нее зависимость и потребуем, чтобы сперва выполнялась задача компиляции.

```
<!-- Jar -->
<target name="jar" description="jar up the project"
depends="compile">
<echo>Jarring the project</echo>
<jar destfile="${dist}/${jarfile}" basedir=".">
<fileset dir="${build}" includes="**/*.class"/>
</jar>
</target>
```

Теперь мы можем запустить наш проект прямо из JAR-файла – осталось указать пути к классам и удостовериться, что мы имеем все необходимые библиотеки. Еще полезнее будет JAR-файла, запускаемый отовсюду: тогда всего одним файлом мы предоставим возможность запуска нашего проекта кому захотим.

Прежде чем создавать JAR-файл-всезнайку, необходимо раздобыть все файлы, которые мы собираемся в нем хранить, и поместить их в тот же каталог: это проще, чем писать Jar-задачу для копирования всех файлов из разных мест. Здесь мы используем каталог **build** для хранения файлов, а каталог **dist** для хранения завершенного JAR-файла. Нижеследующая задача ResrcCopy копирует содержимое каталогов **etc** и **src** в каталог **build**, который будет включен в JAR-файл. Каталоги содержат файлы свойств проекта и его исходный код.

Использование атрибута fileset dir во втором элементе copy означает, что мы копируем весь каталог и его содержимое. Это важно, так как »

Quick tip

Вы можете запустить более чем одну задачу *Ant*, просто указав их одна за одной в командной строке. Например, чтобы очистить проект и затем упаковать его, используйте команду ant clean jar.

Еще один изящный инструмент

Можно предположить, что *Apache Ant* получил такое имя, потому что муравьи очень хорошие строители. Изначально предполагалось, что он заменит make во время сборки движка сервлетов *Apache Tomcat*, для борьбы с проблемами, с которыми автор столкнулся при использовании *make*. Так как *Ant* написан на Java (в отличие от *make*), то он позволяет вести настоящую кросс-платформенную разработку, требуя только совместимую JRE. *Ant* стал стандартным методом для сборки Java-проектов, и вы обнаружите, что большая часть открытых проектов на Java поставляется с файлом сборки для *Ant*.

Установка состоит из разархивирования двоичного дистрибутива в нужный каталог. При компиляции из исходных текстов *Ant* использует интересный подход: сначала частично компилирует себя, а затем использует этот получившийся кусок для компиляции главного двоичного кода *Ant*.

Apache Ant доступен для скачивания с **http://ant.apache.org** по лицензии Apache Software License.

» позволяет установить верную структуру внутри JAR-файла.

```
<!-- ResrcCopy -->
<target name="copyrsrc" description="копируем ресурсы для файла jar">
<echo>Copying resource files for jar</echo>
<copy todir="${build}/etc">
<fileset dir="${etc}"/>
</copy>
<copy todir="${build}/src">
<fileset dir="${src}"/>
</copy>
</target>
```

Далее нам необходимо указать задаче Jar, чтобы мы хотим включить эти новые файлы. Для простоты ссылок и для ясности создадим новый атрибут patternset. Его структура похожа на структуру атрибута path, и включает символы подстановки для типов файлов, которые мы хотим вставить в результирующий JAR-файл.

```
<!-- Jar resource files -->
<patternset id="jar.resources">
<include name="**/*.class"/>
<include name="**/*.properties"/>
<include name="**/*.java"/>
</patternset>
```

Теперь можно внести поправки в задачу Jar и сообщить, что хотим включить эти файлы. Вот первая часть:

```
<!-- Jar -->
<target name="jar" description="jar up the project"
depends="compile, copyrsrc">
<echo>Jarring the project</echo>
<jar destfile="${dist}/${jarfile}" basedir="${build}">
<patternset refid="jar.resources"/>
```

Здесь мы просто изменили атрибут fileset задачи Jar, чтобы он обращался к нашему набору шаблонов. Итого, у нас есть JAR-файл, включающий классы, исходный код и файлы свойств нашего проекта. Но кое-чего не хватает. Одной из причин, по которой мы задумали сделать JAR-файл, была возможность запуска всего проекта из одного файла, без дополнительных библиотек, и здесь возникает небольшая проблема. Мы могли бы включить библиотечные JAR-файлы в готовый patternset, и они отлично бы разместились внутри нашего JAR-файла. Однако Java не позволяет обращаться ко вложенным JAR-файлам. Значит, библиотечные JAR-файлы в определение путей классов нашего JAR-проекта включать нельзя – придется распаковать библиотечные JAR-файлы, затем упаковать заново распакованные классы и их структуру каталогов в наш собственный JAR-файл. Делать это вручную прямо-таки мучительно, и вдобавок во время разработки довольно часто создается другой, новый JAR-файл проекта. Тут-то и приходит на помощь *Ant*: с помощью одного атрибута он может позаботиться для нас обо всем. Вот магическая строка:

```
<zipgroupfileset dir="${libs}" includes="*.jar"/>
```

Как часть задачи Jar, эта строка велит *Ant* включить содержимое всех JAR-файлов в каталог **libs** собственного JAR-файла. Просто, но эффективно.

Последний шаг состоит в создании файла манифеста JAR, с парой атрибутов, которые помогут Java узнать, что делать с JAR-файлом. Ant-задача Jar содержит атрибут manifest, позволяющий их определить.

```
<manifest>
<attribute name="Main-Class" value="${mainclass}"/>
<attribute name="Class-Path" value="."/>
</manifest>
```

Здесь мы можем определить главный класс, выполняемый при запуске JAR-файла (**java -jar <jar_архив>**), и путь к классам, которые JAR-файл должен использовать. Осталось только закрыть Jar-задачу и закончить цель:

```
</jar>
</target>
```

Теперь у нас есть JAR-файл, который может запустить любой обладатель JVM – в некоторых операционных системах это достигается двумя щелчками на нем.

JavaDoc стоит тысячи слов

Любой проект, который мы собираемся сделать открытым, должен иметь включенную документацию Javadoc. Она позволит будущему пользователю просмотреть API и понять, как все работает. *Ant*, разумеется, имеет задачу Javadoc специально для этого.

Вот довольно простая реализация задачи Javadoc:

```
<!-- Javadoc -->
<target name="javadoc" description="create javadocs for the project">
<echo>Creating JavaDoc for project</echo>
<javadoc sourcepath="${src}"
packagenames="org.sturgeon.*"
destdir="${docs}">
</javadoc>
</target>
```

Задача указывает *Ant* создать Javadoc-документацию в каталоге **docs** для исходного кода, расположенного в каталоге **src**, для пакетов, располагающихся ниже *org.sturgeon* (*org.sturgeon.id3*, например). Нужно добавить этот новый каталог **docs** в нашу задачу Init:

```
<mkdir dir="${docs}"/>
```

и задачу Clean:

```
<delete dir="${docs}"/>
```

Нужно также включить документацию в наш JAR-файл, так что припишем пару строк в наш набор шаблонов **jar.resources**, чтобы предусмотреть HTML-файлы Javadoc:

```
<include name="**/*.html"/>
<include name="**/*.gif"/>
```

Чтобы скопировать комплект документации в каталог сборки, включаемый в JAR-файл, надо добавить еще один атрибут copy в задачу ResrcCopy:

Скорая помощь

Eclipse хорошо интегрирован с *Ant*. Чтобы собрать Java-проект *Eclipse* с помощью файла сборки *Ant*, создайте новый проект и выберите Create Java Project From Existing Ant Buildfile. Создастся путь к классам внутри *Eclipse*, и в ваш проект поместятся ваши библиотеки и исходный код. Вы можете также запускать задачи, описанные в файле сборки, прямо из *Eclipse*.

Пробуем озадачиться

Существует много сторонних задач для *Ant*, которые вы можете попробовать. Вот список самых интересных:

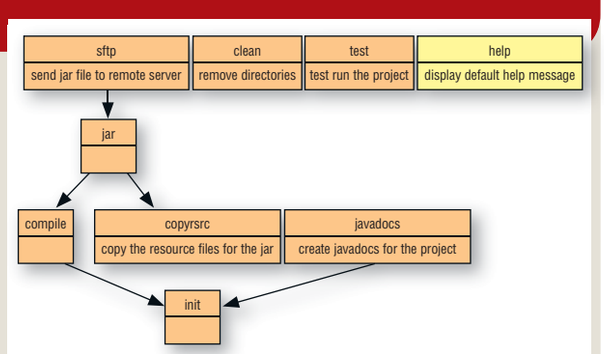
» *Linguine Maps* (www.softwaresecretweapons.com/jspwiki/ Wiki.jsp?page=LinguineMaps) Автоматически представит ваш файл сборки в виде диаграммы.

» *Ant2Svg* (www.spiritedsw.com/ant2svg) Создаст простое графическое представление вашего файла сборки в формате SVG.

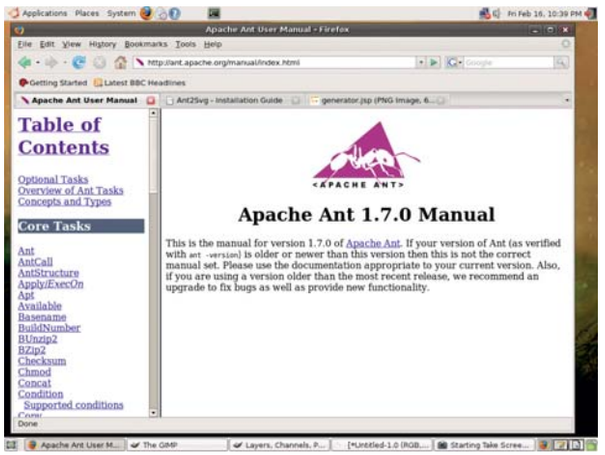
» *Jakarta Ant Image* (www.mullassery.com/software/ANT)

Может проводить преобразования вроде поворота изображений, определенных в файле сборки.

» *ImportScrubber* (http://importscrubber.sourceforge.net) Подчистит все нежелательные выражения импорта.



› ***Linguine Maps* нарисует представление файла сборки Java.**



› Это руководство для Ant: оно прямо плачет по задаче Roomba...

```
<copy todir ="${build}/docs">
<fileset dir="${docs}"/>
</copy>
```

Готово: теперь у нас есть JAR-файл, который не только является исполняемым, но и содержит исходный код и документацию Javadoc.

Упакован и отгружен

Если бы этот JAR-файл был EJB, разве не славно было бы, чтобы он автоматически развертывался на EJB-сервере? Положим, что наш удаленный сервер позволяет только безопасные соединения, и что для передачи JAR-файла требуется безопасный FTP. Как обычно, *Ant* припас задачу, способную это сделать, и называется она Scp. Однако, так как это одна из опциональных задач *Ant*, нужно добавить внешние библиотеки для ее активации. Расположение внешних библиотек описано на странице библиотек зависимостей в руководстве *Ant*; можно видеть, что для задачи Scp требуется файл **jsch.jar** со страницы **www.jcraft.com/jsch/index.html**. Скачав библиотечный файл, скопируйте его в каталог библиотек *Ant* (**ANT_HOME/lib**).

Далее нам необходимо добавить несколько переменных в файл **local.properties**: они укажут *Ant*, куда положить файл.

```
user=art
passwd=password
backendserver=www.sturgeon.org
remotedir=/opt/id3
```

Теперь мы можем добавить новую безопасную задачу FTP (sftp) в наш существующий файл сборки. Мы добавим в зависимости задачу Jar, так нам нужно, чтобы файл существовал до того, как мы сможем его куда-либо отослать!

```
we need this file to exist before we can send it anywhere!
<!-- sftp -->
<target name="sftp" description="send jar file to remote server"
depends="jar">
<echo>Copying file to remote directory</echo>
<scp trust="yes" todir="${user}:${passwd}@${backendserver}:${
remotedir}">
<fileset dir="${dist}">
<include name="${jarfile}"/>
</fileset>
</scp>
</target>
```

Как вы можете видеть, эта задача использует тот же подход, что и другие наши задачи: за начальным объявлением следует атрибут fileset, определяющий файлы, которые мы хотим использовать в задаче. Один из главных атрибутов задачи – trust, приказывающий Scp всегда доверять неизвестным узлам. Без него потребовалось бы определить файл **knownhosts** с доверенными узлами. Сконструируем атрибут todir используя следующий формат: user:password@server:directory. Внутри набора файлов мы просто укажем JAR-файл, который хотим переслать.

Полезный трюк, если вы посылаете JAR-файл на *Jboss* (или аналогичный сервер приложений) – использовать команду sleep. Если вы удаляете старый JAR-файл EJB и затем сразу же загружаете новый, то *JBoss*, возможно, не успеет удалить старый файл EJB перед тем, как получить новый – что может привести к недоразумениям. Чтобы этого избежать, мы можем просто использовать задачу Sleep между удалением старого EJB и копированием нового файла, чтобы дать шанс *JBoss* завершить удаление:

```
<sleep seconds="5"/>
```

Можно также определить время в минутах и часах, если ваш сервер особо важных приложений работает на ветхом оборудовании.

В описанной задаче мы посылаем файл через безопасный FTP. При желании использовать старый добрый FTP-стандарт мы могли бы использовать задачу FTP. Так как это тоже опциональная задача, пришлось бы установить общую сетевую внешнюю библиотеку *Jakarta*, которую требует задача *Ant* FTP для своего функционирования.

Я не знаю слова «помощь»

В качестве финального аккорда, создадим задачу Help (Помощь) – пусть выводит список всех возможных задач на консоль. Хотя это довольно просто, но очень полезно, особенно если вас интересуют промежуточные задачи, запускаемые из-за зависимостей.

```
<!-- Help -->
<target name="help" description="display default help message">
<echo>help - display this message</echo>
<echo>init - Initialise the directory structure</echo>
<echo>clean - Clean directory structure</echo>
<echo>compile - Compile the source code</echo>
<echo>test - Test run the project</echo>
<echo>jar - Create project jar file</echo>
<echo>javadoc - Create JavaDocs for the project</echo>
<echo>sftp - Send the Jar file to the remote directory</echo>
</target>
```

Было бы мило с нашей стороны предоставить помощь по умолчанию: если кто-то вызовет *Ant* с нашим файлом проекта, не указав задачу, то увидит справку. Для этого достаточно определить свойство default у атрибута project:

```
<project name="id3" default="help" basedir=".">
```

Все описанные задачи можно было выполнить и вручную, с помощью стандартных инструментов командной строки, типа *cp* и *mv*. Но стоит помнить, что написав эти задачи, вы можете вызвать их сколь угодно раз. Если по-умному использовать зависимости, сложный проект можно разбить на ряд маленьких понятных команд. Так как хлопот при создании нового проекта немало, то возможность запуска одной задачи *Ant*, охватывающей множество разных аспектов проекта, является манной небесной. LXF

Скорая помощь

Если вы не знаете, какие атрибуты использовать, посмотрите руководство по *Ant* (картинка слева). Оно доступно в сети на странице *Ant*, а также поставляется вместе с дистрибутивом *Ant*; в нем есть описание всех атрибутов, а также хорошие примеры – его стоит почитать.

» **Через месяц** «Красота– это функция»: изучим тэг <canvas> из Firefox

Программирование GNOME —

ЧАСТЬ 6 *GTK+* – это не только инструментарий, это еще и основа сразу для двух рабочих сред! Настало время разобраться с GNOME – **Андрей Боровский** показывает, как создать свое первое GNOME-приложение.

GNOME и *GTK+* соотносятся между собой примерно так же, как KDE и *Qt*, с той разницей, что разработчики открытых сообществ GNOME и *GTK+* всегда лучше взаимодействовали между собой, чем разработчики «соборного» *Qt* и «базарного» KDE.

Если вы стоите перед выбором – использовать ли в своей программе только визуальные компоненты *GTK+*, или же добавить к ним расширения GNOME, соображения в пользу каждого варианта будут примерно такими же, что и при выборе между «чистым» *Qt* и KDE. Компоненты GNOME могут сделать больше (правда, ненамного), чем компоненты *GTK+*, но GNOME – это не просто набор виджетов, а оболочка, и программа, использующая возможности GNOME, сможет работать только на том компьютере, где установлена эта оболочка. В то же время программы, основанные на *GTK+*, могут быть перенесены и в такие среды, где гномы сроду не водились – например, на платформу Microsoft Windows. Впрочем, говоря о том, что нет GNOME для Windows, я не совсем прав: его портирование выполняется в рамках проекта *Cygwin*. Однако нельзя сказать, чтобы CyGNOME был популярен среди пользователей Windows, так что и желающих установить его ради одной вашей программы найдется немного.

В то время как в тандеме *Qt*/KDE практически каждому компоненту *Qt* соответствует свой компонент KDE, компоненты *GTK+* и GNOME практически не дублируют друг друга. К богатому набору *GTK+* GNOME добавляет в основном те компоненты, которые требуются для написания программ, взаимодействующих с оболочкой GNOME. Среда GNOME включает в себя несколько разделяемых библиотек, с которыми компонуются исполняемые файлы приложений GNOME. Важнейшими из этих библиотек являются две – *libgnome* (известная в документации как GNOME Library) и *libgnomeui* (в документации она значится под именем GNOME UI Library). Библиотека *libgnome* экспортирует базовые функции, необходимые для инициализации GNOME-программы, конфигурации программ GNOME и интернационализации. Кроме того, библиотека *libgnome* предоставляет в распоряжение программиста несколько функций для решения таких задач, как, например, отображение встроенной справки и работа со звуком. Библиотека *libgnomeui* экспортирует функции, обеспечивающие работу дополнительных визуальных элементов GNOME.

Наша первая программа для GNOME (файл **openurl.c**) не так уж сильно отличается от простой программы *GTK+*:

› **Рис. 1.**

```
#include <gnome.h>

void button_clicked(GtkWidget * button, gpointer data)
{
    GtkEntry * entry = data;
    GError * error = NULL;
    const char * url = gtk_entry_get_text(entry);
    if (!gnome_url_show(url, &error)) {
      g_print("%s\n", error->message);
      g_error_free(error);
    }
}

gint delete_event(GtkWidget *widget, GdkEvent *event, gpointer data)
{
    gtk_main_quit();
    return FALSE;
}

int main(int argc, char * argv[])
{
    GnomeProgram * gnome_prog;
    GtkWidget * mainwnd;
    GtkWidget * hbox;
    GtkWidget * button;
    GtkWidget * entry;
    gnome_prog = gnome_program_init("openurl", "0.1", LIBGNOMEUI_
MODULE,
        argc, argv, NULL, NULL);
    mainwnd = gnome_app_new ("openurl", "Открыть URL");
    gtk_signal_connect (GTK_OBJECT (mainwnd), "delete_event",
                GTK_SIGNAL_FUNC(delete_event), NULL);
    hbox = gtk_hbox_new (FALSE,5);
    entry = gtk_entry_new();
    gtk_box_pack_start (GTK_BOX(hbox), entry, TRUE, TRUE, 0);
    button = gtk_button_new_with_label("Открыть");
    gtk_signal_connect (GTK_OBJECT (button), "clicked",
      GTK_SIGNAL_FUNC (button_clicked), entry);
    gtk_box_pack_start (GTK_BOX(hbox), button, FALSE, FALSE, 0);
    gnome_app_set_contents (GNOME_APP (mainwnd), hbox);
    gtk_widget_show_all(mainwnd);
    gtk_main ();
    return 0;
}
```

Месяц назад Мы завершили изучение «чистого» *GTK+* обсуждением новых возможностей *Glade 3.x*.

первые шаги

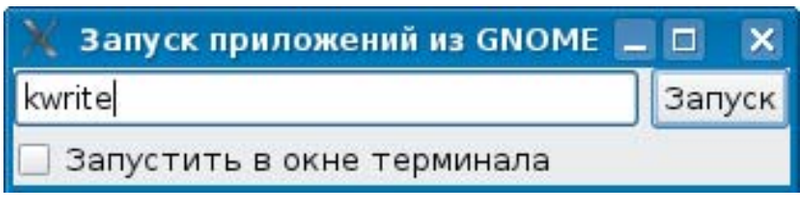
Прежде чем разбирать исходный текст программы, полезно узнать, что она делает. Окно программы *openurl* (**Рис. 1**) содержит строку ввода и кнопку. Если ввести в строке адрес URL, а затем щелкнуть кнопку мышью, программа запустит соответствующий внешний интернет-клиент и передаст ему введенный URL. Под «соответствующим Интернет-клиентом» в данном случае понимается клиентская программа, которая назначена в вашей системе для обработки указанного в URL интернет-протокола (HTTP, FTP и т.д.) по умолчанию. Информацию о том, какую именно программу нужно запустить для обработки той или иной ссылки URL, программа *openurl* получает из настроек системы, так что если программа не может открыть ссылку какого-то типа, проверьте, назначен ли для соответствующего протокола клиент по умолчанию. Пошаговый разбор программы *openurl* мы начнем с заголовочного файла. Хотя наша программа использует функции библиотек *libgnome*, *libgnomeui* и, конечно, *GTK+*, нам достаточно включить в исходный текст один заголовочный файл – **gnome.h**. В результате станут доступны прототипы функций всех перечисленных интерфейсов. Пропустим пока обработчики сигналов и рассмотрим функцию main(). Работа главной функции программы начинается с вызова функции gnome_program_init(). Она инициализирует библиотеки GNOME (и *GTK+*), с которыми будет взаимодействовать наша программа. Первым аргументом функции gnome_program_init() является идентификатор приложения – это может быть просто строка с именем исполняемого файла программы. Вторым аргументом должна быть строка с версией приложения. Помимо прочего, функция gnome_program_init() загружает информацию о модулях GNOME, требуемых приложению. Для того, чтобы сообщить системе какие модули нам нужны, достаточно указать в качестве третьего параметра макрос-константу Libgnomeui_MODULE. В качестве четвертого и пятого параметров функции gnome_program_init() передаются параметры argc и argv, полученные функцией main(). Остальные параметры gnome_program_init() (а их может быть много) связаны с обработкой ключей командной строки. Мы не собираемся запускать нашу программу с какими-либо дополнительными ключами и просто передаем два значения **NULL** для того, чтобы функция gnome_program_init() была довольна. В предыдущих версиях GNOME для инициализации программы использовалась функция gnome_init(). И хотя эта функция до сих пор присутствует в интерфейсе GNOME, использовать ее в новых программах категорически не рекомендуется. По моим наблюдением, использование gnome_init() с последними версиями GNOME может нарушить стабильность работы всей графической среды. Функция gnome_program_init() возвращает указатель на объект GnomeProgram. Он содержит много полезной информации о запускаемой GNOME-программе, но обращаться к нему мы пока не будем.

После того, как все нужные нам библиотеки GNOME инициализированы, мы можем приступить к созданию интерфейса программы. Конструирование интерфейса мы начнем с главного окна, которое создается функцией gnome_app_new(). Первым аргументом этой функции должно быть имя программы, вторым аргументом – заголовок главного окна. Функция возвращает указатель на объект GnomeApp, который, вопреки своему названию, является всего лишь улучшенным вариантом (и потомком) объекта GtkWindow. Назначение обработчика события delete_event, создание контейнера для горизонтальной упаковки дочерних элементов, создание и упаковка самих элементов GtkEntry (строка ввода) и GtkButton (кнопка), так же как и назначение обработчика сигнала clicked кнопки, должны быть вам хорошо знакомы, так что мы их пропустим. Функция gnome_app_set_contents() позволяет указать визуальный элемент, который будет отвечать за содержимое рабочей области главного окна программы. Первым аргументом функции gnome_app_set_contents() должен быть указатель на окно GnomeApp, вторым – указатель на его дочерний элемент, управляющий рабочей областью(в качестве такового в нашей программе выступает объект-контейнер hbox).

Функция gtk_widget_show_all() является частью интерфейса *GTK+*, но мы с ней раньше не встречались, поэтому опишем ее подробнее. До сих пор мы делали каждый элемент управления программы видимым с помощью отдельного вызова gtk_widget_show(). Функция gtk_widget_show_all() позволяет «придать видимость» всем визуальным элементам программы за один вызов. Цикл обработки сообщений программы GNOME запускается с помощью функции gtk_main(), предоставляемой интерфейсом *GTK+*.

Перейдем к обработчику сигнала clicked кнопки button – функции button_clicked(). В качестве дополнительного параметра функции-обработчику передается указатель на объект GtkEntry, из которого он должен получить текст адреса URL. Вся «магическая работа» по запуску внешней программы для обработки URL выполняется функцией gnome_url_show(). Ее первым параметром должна быть строка URL, второй аргумент представляет собой указатель на указатель на структуру GError. Если в процессе обработки URL функцией gnome_url_show() возникла ошибка, функция создаст экземпляр структуры GError, и вернет указатель на него во втором параметре. При этом функция также вернет значение **FALSE** как результат вызова. Мы проверяем значение, возвращаемое gnome_url_show(), и в случае возникновения ошибки распечатываем сообщение о ней (текст сообщения содержится в поле message структуры GError). Поскольку при возникновении ошибки каждый раз создается новый экземпляр структуры GError, мы должны удалять его с помощью функции g_error_free(), чтобы не заполнять оперативную память мусором. Обратите внимание, что функция gnome_url_show() следит за тем, чтобы переданная ей переменная-указатель на объект GError имела значение **NULL**. В противном случае функция может, ни много ни мало, досрочно завершить работу программы.

Прежде чем вы начнете проверять, как работает программа openurl, необходимо сказать несколько слов о том, как функция gnome_url_show() обрабатывает переданные ей URL-адреса. Для того, чтобы выбрать внешнюю программу для открытия того или иного URL, функция должна «знать» интернет-протокол, который он использует. Информацию о протоколе функция получает из префикса URL (**http://**, **ftp://** и т.д.), поэтому, вводя URL в строке ввода программы, вы обязательно должны указывать префикс (если вы его не укажете, gnome_url_show() вернет сообщение об ошибке). Отметим также, что ошибки, о которых может сообщить функция gnome_url_show(), относятся исключительно к синтаксису адреса URL, но не к ресурсу, на который она ссылается. Как только функция находит приложение, способное открыть данный URL, она запускает ее, передавая URL в качестве параметра командной строки и возвращает управление нашей программе. Вообще говоря, у нас нет возможности проследить, чем закончилась »



› Рис. 2.

» работа программы, которую запустила функция gnome_url_show().

Компиляция и сборка программы для GNOME требуют дополнительных приготовлений. Лучше всего воспользоваться простым make-файлом. Рассмотрим фрагмент из make-файла, который вы найдете на диске (он собирает все программы-примеры для этой статьи, а для того, чтобы собрать программу *openurl*, вы можете просто скомандовать “make openurl”):

```
CFLAGS =-g -Wall `pkg-config --cflags libgnome-2.0 libgnomeui-2.0`
LDFLAGS=`pkg-config --libs libgnome-2.0 libgnomeui-2.0`
```

Для генерации ключей компилятора и компоновщика мы вызываем утилиту *pkg-config*, указав ей пакеты *libgnome-2.0* и *libgnomeui-2.0*. Теперь программу можно и скомпилировать.

Открыть в терминале?

Запуск внешних Интернет-клиентов – это далеко не все, на что способны программы GNOME. В качестве примера более широких возможностей мы рассмотрим приложение, способное запустить произвольно заданную программу. Наша вторая программа GNOME (ее исходный текст вы найдете в файле **gnomeexec.c**) похожа на первую (**Рис. 2**). Введите имя программы, которую вы хотите запустить, и нажмите кнопку Запуск. Если вы собираетесь работать с консольной программой, можете установить флажок Запустить в окне терминала (в этом случае при запуске программы будет открыто окно терминала, используемого GNOME по умолчанию).

Главная проблема, с которой сталкивается программист, пишущий Unix-программу, которая должна запускать другие программы – это проблема файловых дескрипторов. По умолчанию, дочерний процесс наследует все дескрипторы, открытые родительским процессом. В графической системе GNOME, в которой каждая программа открывает множество служебных дескрипторов, в том числе для связи с сервером X и CORBA, разделение этих дескрипторов с новым процессом может вызвать проблемы у обеих программ. Запуская новую программу, программист должен быть уверен, что она не унаследует дескрипторы, которые должны принадлежать исключительно родительской программе. Если вы думаете, что далее я приведу пример закрытия дескрипторов (как в статьях по программированию Unix API), то ошибаетесь. Дело в том, что интерфейс программирования GNOME предоставляет в распоряжение программиста целое семейство функций, предназначенных для запуска из программы GNOME других программ. Все проблемы, связанные с дескрипторами, эти функции решают за нас.

Всего в нашем распоряжении находится восемь функций интерфейса GNOME, позволяющих запустить внешнюю программу. Мы воспользуемся двумя – gnome_execute_shell() и gnome_execute_terminal_shell(). Эти функции позволяют запускать внешние программы с теми переменными окружения, которые установлены в оболочке, используемой пользователем по умолчанию. Функция gnome_execute_shell() просто запускает программу на выполнение, а функция gnome_execute_terminal_shell() сначала открывает окно терминала и запускает программу в нем. Обе функции принимают два аргумента – рабочую директорию программы и строку запуска. Если в качестве первого аргумента функциям передать значение **NULL**, новое приложение унаследует рабочий каталог программы-родителя.

Рассмотрим обработчик сигнала clicked, определенный в программе *gnomeexec*. Именно в нем и выполняется запуск внешней программы:

```
GtkWidget * entry;
GtkWidget * check_button;

void button_clicked(GtkWidget * button, gpointer data)
{
    const char * cmdline = gtk_entry_get_text(GTK_ENTRY(entry));
    if (gtk_toggle_button_get_active(GTK_TOGGLE_BUTTON(check_
button)))
                gnome_execute_terminal_shell(NULL, cmdline);
        else
                gnome_execute_shell(NULL, cmdline);
}
```

В программе *openurl* обработчику требовались данные только одного элемента управления – строки ввода. В программе *gnomeexec* обработчик должен, помимо строки запуска программы, получить данные о состоянии флажка, который определяет, следует ли запускать программу в окне терминала. В качестве строки ввода мы используем все тот же объект GtkEntry. Кнопка-флажок реализована в *GTK+* с помощью объекта GtkCheckButton. Таким образом, обработчик сигнала clicked должен получить информацию сразу о двух объектах. Для того, чтобы они были доступны обработчику, мы объявляем переменные entry и check_button глобально, в области видимости обработчика и функции main().

Из объекта entry обработчик извлекает строку запуска программы. Какая именно из функций запуска внешней программы будет использована, зависит от того, установлен ли флажок check_button. Его состояние проверяется с помощью функции gtk_toggle_button_get_active() (объект GtkCheckButton является потомком объекта GtkToggleButton, который служит предком всех кнопок-переключателей). Если функция gtk_toggle_button_get_active() возвращает TRUE, значит, флажок установлен и мы запускаем внешнюю программу в окне терминала с помощью функции gnome_execute_terminal_shell(). Если функция gtk_toggle_button_get_active() возвращает значение **FALSE**, для запуска программы используется функция gnome_execute_shell().

В программе *gnomeexec* есть еще несколько строчек кода, заслуживающих вашего внимания. Мы привыкли к тому, что если в окне программы есть «главная» кнопка, выполняющая некое действие, то нажатие клавиши Enter равносильно щелчку мышью по этой кнопке. Работая с программой *openurl*, вы наверняка заметили, что эта программа игнорирует клавишу Enter, так что кнопку Открыть приходится щелкать мышью. В окне, где сначала нужно ввести текст с клавиатуры, необходимость переключаться на мышь для щелчка по кнопке неудобна вдвойне. В программе *gnomeexec* мы исправим этот недостаток, но прежде – небольшое лирическое отступление. Мы любим программирование за то, что оно позволяет нам реализовать наши творческие замыслы, и ненавидим его за разные неожиданные препятствия, которые оно возводит на нашем пути. Как заставить кнопку реагировать на нажатие Enter? Знатоки графических интерфейсов могут ожидать, что существует какая-нибудь функция gtk_button_set_default(), которая делает заданную кнопку «кнопкой по умолчанию». Ничего подобного в *GTK+ API* нет. Чтобы заставить кнопку реагировать на клавишу Enter, нам придется вызвать две функции и один макрос. Прежде всего, если в окне есть объект GtkEntry (а в окне *gnomeexec* он есть), по умолчанию он перехватывает все события, связанные с клавиатурой. Для того, чтобы объект GtkEntry мог передать обработку нажатия на клавишу Enter другим элементам управления окна, мы должны вызвать функцию gtk_entry_set_activates_default(). Далее, оказывается, что кнопка GtkButton по умолчанию не относится к числу объектов, способных (простите за каламбур) обрабатывать нажатие Enter по умолчанию. Мы, однако, можем наделить кнопку GtkButton этой способностью, если вызовем макрос GTK_WIDGET_SET_FLAGS() с константой GTK_CAN_DEFAULT:

```
GTK_WIDGET_SET_FLAGS (button, GTK_CAN_DEFAULT);
```

Макрос GTK_WIDGET_SET_FLAGS() позволяет устанавливать различные флаги, влияющие на свойства и поведение визуальных элементов. Только теперь мы можем вызвать функцию gtk_widget_grab_default(), которая заставит кнопку реагировать на нажатие Enter как на щелчок мышью. Справедливости ради стоит отметить, что при использовании окна, создаваемого объектом GtkDialog, определение «кнопок

по умолчанию» упрощается. После того, как поведение кнопки приведено в соответствие с правилами хорошего интерфейса, мы можем собрать программу, скомандовав

```
make gnomeexec
```

А как это будет по-русски?

Интернационализация приложений GNOME выполняется почти так же, как и приложений *GTK+* (в связи с чем рекомендую перечитать LXF87/88), однако есть и небольшие отличия. В процессе интернационализации приложений *GTK+* нам требовалось самим определить несколько макросов. В заголовочных файлах GNOME эти макросы уже определены, и повторное их определение приведет к выдаче компилятором предупреждающих сообщений. В качестве примера выполним интернационализацию программы *gnomeexec* (этот вариант программы вы найдете в файле **gnomeexec-i18n.c**). Никаких дополнительных заголовочных файлов в текст программы включать не требуется – достаточно уже имеющегося **gnome.h**. Наши действия по подготовке интерфейса программы к переводу на другие языки сводятся к трем шагам. Во-первых, в функцию main() мы добавляем вызовы трех макросов:

```
bindtextdomain (GETTEXT_PACKAGE, LOCALEDIR);
bind_textdomain_codeset (GETTEXT_PACKAGE, “UTF-8”);
textdomain (GETTEXT_PACKAGE);
```

То же самое мы делали в процессе интернационализации программы *GTK+*. Макросы располагаются в самом начале функции main(), еще до вызова функции gnome_program_init().

Второй шаг заключается в пометке всех строк, предназначенных для перевода, макросом _(). В *GTK+* мы определяли этот макрос сами, в GNOME он уже определен.

Нашим третьим шагом должно быть определение констант-макросов GETTEXT_PACKAGE и LOCALEDIR. Поскольку для компиляции программы мы используем make-файлы, будет удобно разместить объявления этих макросов в нем. В результате исходный текст нашей программы вообще не будет содержать никаких определений макросов. В make-файл для сборки программы *gnomeexec-i18n* мы добавим строку

```
CFLAGS =-g -Wall `pkg-config --cflags libgnome-2.0 libgnomeui-2.0` \\
-DLOCALEDIR=\””./locale”\” -DGETTEXT_PACKAGE=\””gnomeexec-i18n”\” \\
-DENABLE_NLS
```

Помимо двух вышеуказанных констант мы включаем директиву условной компиляции ENABLE_NLS, которая делает доступными все макросы и функции интернационализации, объявленные в заголовочных файлах, включенных в **gnome.h**. Теперь текст интерфейса программы *gnomeexec-i18n* готов к переводу. Остальные действия, связанные с представлением его интерфейса на других языках, выполняются точно так же как и при работе с *GTK+*. С помощью утилиты *xgettext* мы создаем каталог строк, предназначенных для перевода, и переводим строки из каталога, используя любую подходящую утилиту (например, *KBabel*), затем, при помощи утилиты *msgfmt*, компилируем переведенный каталог в двоичный файл перевода и размещаем его в директории, указанной в константе LOCALEDIR. Серьезные приложения GNOME, предназначенные для всех пользователей системы, размещают свои файлы перевода в директории **/opt/gnome/share/locale**, мы же не претендуем на такую честь и помещаем файл перевода нашей программы в локальном каталоге.

Я признаю, что программы из этой статьи отличались довольно незамысловатым интерфейсом. Все дело в том, что мне лень вручную громоздить контейнеры друг на друга (даже если эти контейнеры – всего лишь объекты *GTK+*). Следующая статья этой серии расскажет вам, как создавать роскошные интерфейсы GNOME с помощью программ *Glade* и *Anjuta*. LXF

» **Через месяц** Визуальное проектирование интерфейсов GNOME – и загадочная Anjuta.

С окнами на «ты»

ЧАСТЬ 12 В заключительной статье цикла о программировании для Unix **Андрей Боровский** расскажет об использовании цветов и поддержке мыши... в консоли!

Мы продолжаем знакомство с ncurses. В прошлый раз мы научились создавать окна. На этом уроке мы рассмотрим другие важные возможности ncurses, такие, как управление цветом и поддержка мыши.

Управление цветом

Принципы работы с цветом в ncurses могут оказаться неожиданными для тех, кто привык работать с цветами в растровых графических системах (и для тех, кто имеет опыт работы с текстовым режимом DOS/Windows). Библиотека *ncurses* инициализирует восемь базовых цветов: черный, красный, зеленый, желтый, синий (blue), ярко-красный (magenta), голубой (cyan) и белый (базовыми называются цвета с обычным уровнем яркости). Поскольку к каждому базовому цвету можно применить атрибут повышенной яркости A_BOLD, всего мы получаем 16 цветов (в результате применения атрибута A_BOLD к черному цвету получается темно-серый цвет). Базовым цветам соответствуют константы COLOR_BLACK, COLOR_RED, COLOR_GREEN, COLOR_YELLLOW, COLOR_BLUE, COLOR_MAGENTA, COLOR_CYAN и COLOR_WHITE (для черного, красного, зеленого, желтого, синего, ярко-красного, голубого и белого цветов соответственно). Следует отметить, что фактические цвета в окне терминала зависят, прежде всего, от настроек самого терминала. Например, базовый желтый цвет (COLOR_YELLLOW) будет выглядеть скорее как коричневый, а для того, чтобы он стал собственно желтым, ему необходимо придать атрибут повышенной яркости. Библиотека *ncurses* позволяет определять собственные цвета с помощью функции init_color, но эта возможность поддерживается не всеми консолями. Позволяет ли консоль определять собственные цвета, можно выяснить с помощью функции can_change_color(). Цвета в *ncurses* объединяются в пары – цвет символов (foreground) и цвет фона (background). Перед тем как печатать цветной текст, необходимо определить соответствующую цветовую пару и установить ее в качестве атрибута текста (так же, как устанавливается атрибут мигания или подчеркивания). Изменить цвет фона или символов независимо друг от друга нельзя, необходимо определять новую пару. Система управления цветами ncurses инициализирует две переменные – COLORS (количество базовых цветов) и COLOR_PAIRS (максимальное количество цветовых пар, которые можно определить одновременно). При работе с терминалом *konsole* эти переменные принимают значения 8 и 64 соответственно.

Рассмотрим управление цветом на примере программы *cursedcolors* (на диске – файл **cursedcolors.c**)

```
#include <termios.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <stdlib.h>
#include <curses.h>

void sig_winch(int signo)
{
    struct winsize size;
    ioctl(fileno(stdout), TIOCGWINSZ, (char *) &size);
    resizeterm(size.ws_row, size.ws_col);
}

int main(int argc, char ** argv)
{
    WINDOW * wnd;
    WINDOW * subwnd;
    initscr();
    signal(SIGWINCH, sig_winch);
    curs_set(FALSE);
    start_color();
    refresh();
    init_pair(1, COLOR_BLUE, COLOR_GREEN);
    init_pair(2, COLOR_YELLOW, COLOR_BLUE);
    wnd = newwin(5, 18, 2, 4);
    wattron(wnd, COLOR_PAIR(1));
    box(wnd, 'l', '-');
    subwnd = derwin(wnd, 3, 16, 1, 1);
    wbkgd(subwnd, COLOR_PAIR(2));
    wattron(subwnd, A_BOLD);
    wprintw(subwnd, "Hello, brave new curses world!\n");
    wrefresh(subwnd);
    wrefresh(wnd);
    delwin(subwnd);
    delwin(wnd);
    wmove(stdscr, 8, 1);
    printw("Press any key to continue...");
    refresh();
    getch();
    endwin();
    exit(EXIT_SUCCESS);
}
```

Эта программа основана на программе *cursedwindows* из предыдущей статьи, так что многие ее части должны быть вам знакомы. Функция start_color() инициализирует управление цветом ncurses. Остальные функции, связанные с цветом, можно использовать только после вызова start_color(). Новые цветовые пары создаются с помо-

Месяц назад Мы познакомились с ncurses и научились создавать свои окна.

щью функции init_pair(). Первым параметром init_pair() должен быть один из допустимых номеров пары (от 1 до COLOR_PAIRS-1). Вторым параметром функции init_pair() должна быть константа, обозначающая базовый цвет символа, а третьим – константа, обозначающая базовый цвет фона. Цветовая пара с номером 0 определена в *ncurses* как «белый на черном», и изменить ее нельзя. Мы создаем две пары цветов – «синие символы на зеленом фоне» под номером 1 и желтые символы на синем фоне (любимое сочетание цветов небезызвестного Питера Нортона) под номером 2. Номер цветовой пары служит ее идентификатором. Для того чтобы сделать выбранную цветовую пару атрибутом выводимого текста, необходимо установить с помощью функции attron/wattron атрибут COLOR_PAIR(X), где X – номер цветовой пары. Атрибут COLOR_PAIR(X) можно комбинировать с другими, например, с атрибутом A_BOLD, который влияет на яркость цвета символов (но не на яркость цвета фона). Для того чтобы изменить яркость фона, необходимо скомбинировать этот атрибут с A_REVERSE.

Вызов функции

```
wattron(wnd, COLOR_PAIR(1));
```

устанавливает цвет фона и символов (цветовую пару 1) для «внешнего» окна wnd, содержащего рамку. Теперь функция box() напечатает символы рамки с учетом заданных атрибутов цвета. Функция wbkgd() позволяет нам заполнить структуру данных, соответствующую массиву символов окна, различными атрибутами текста. Вызов

```
wbkgd(subwnd, COLOR_PAIR(2));
```

заполняет окно subwnd фоновым цветом из цветовой пары 2 и устанавливает соответствующий цвет символов в окне. Помимо атрибута COLOR_PAIR(), этой функции можно передавать все те же комбинации атрибутов, что и wattron(). Атрибуты затем будут применены к тексту, выводимому в окне по умолчанию. Для того чтобы сделать цвет шрифта в окне subwnd ярким, мы вызываем функцию wattron() с атрибутом A_BOLD. Заметьте, что в функции wattron(), вызванной для окна subwnd, мы не указываем цветовую пару, поскольку в этом нет необходимости. Функция wbkgd() уже заполнила символьный массив окна subwnd нужными атрибутами цвета, и нам остается только указать атрибут яркости. В принципе, мы могли бы обойтись и без wattron(), если бы вызов wbkgd() выглядел так:

```
wbkgd(subwnd, COLOR_PAIR(2)|A_BOLD);
```

Теперь мы можем распечатать текст в окне, что и делается с помощью функции wprintw(). Для того чтобы символы, напечатанные в окне, стали видимыми, необходимо вызвать функцию wrefresh(). Теперь окно с текстом и обрамляющая его рамка сияют разными цветами (**Рис. 1**). Вы могли заметить, что если в обычном режиме окно терминала было, например, белым, то во время работы программы *cursedcolors* оно становится черным. Это происходит потому, что по умолчанию при инициализации цвета окно stdscr заполняется атрибутами цветовой пары 0, как если бы была вызвана функция

```
wbkgd(stdscr, COLOR_PAIR(0));
```

В результате надпись *"Press any key to continue..."*, которую мы печатаем в окне stdscr, выводится белым шрифтом на черном фоне.

При работе с цветом в ncurses следует помнить о том, что массивы данных окон хранят только номера цветовых пар, применяемых к каждой ячейке, а не сами значения цветов. Из этого следует, что цвета уже напечатанного текста зависят от определения цветовых пар. Допустим, вы определили цветовую пару 1 как «желтый на синем» и напечатали какой-нибудь текст, используя эту пару в качестве атрибута. Если затем вы переопределите цветовую пару 1 как «красный на белом», цвета шрифта и фона в уже напечатанном тексте изменятся соответственно новому определению цветовой пары.

Ввод данных в окнах

С одной из функций ввода данных – getch(), мы уже познакомились. Мы также знаем, что этой функции соответствует «оконная» функция wgetch(). Обе они позволяют считывать отдельные символы. В отличие от них, семейство функций getstr()/getnstr/wgetstr()/wgetnstr() позволяет считывать целые строки. Буква n перед str в именах функций свидетельствует о том, что эти варианты функций позволяют указать максимальную длину строки-буфера и, тем самым, избежать его переполнения при вводе. Программа *cursedinput* (ее исходные тексты вы найдете в файле **cursedinput.c**) позволяет пользователю вводить строку текста и, затем, распечатывает введенную строку.

```
#include <termios.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <stdlib.h>
#include <curses.h>

#define MAX_NAME_LEN 15
void sig_winch(int signo)
{
    struct winsize size;
    ioctl(fileno(stdout), TIOCGWINSZ, (char *) &size);
    resizeterm(size.ws_row, size.ws_col);
}

int main(int argc, char ** argv)
{
    WINDOW * wnd;
    char name[MAX_NAME_LEN + 1];
    initscr();
    signal(SIGWINCH, sig_winch);
    curs_set(TRUE);
    start_color();
    refresh();
    init_pair(1, COLOR_YELLOW, COLOR_BLUE);
    wnd = newwin(5, 23, 2, 2);
    wbkgd(wnd, COLOR_PAIR(1));
    wattron(wnd, A_BOLD);
    wprintw(wnd, "Enter your name...\n");
    wgetnstr(wnd, name, MAX_NAME_LEN);
    name[MAX_NAME_LEN] = 0;
    wprintw(wnd, "Hello, %s!", name);
    wrefresh(wnd);
    delwin(wnd);
    curs_set(FALSE);
    move(8, 4);
    printw("Press any key to continue...");
    refresh();
    getch();
    endwin();
    exit(EXIT_SUCCESS);
}
```

Поскольку в программе *cursedinput* пользователь должен вводить данные, нам удобно сделать курсор видимым, что мы и делаем с помощью вызова curs_set(TRUE). Собственно ввод строки выполняется с помощью wgetnstr(). Первый параметр функции – идентификатор окна, в котором вводятся данные (то есть отображаются вводимые символы и курсор), второй параметр – строка-буфер, в которую записываются введенные символы, а третьим аргументом является длина буфера. При работе с программой вы увидите, что нельзя ввести число символов, превышающее MAX_NAME_LEN. Перед выводом строки *"Press any key to continue..."* мы снова прячем курсор.

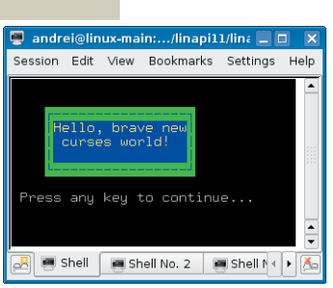


Рис. 1 Разноцветные окна в *ncurses*.

Режим работы терминала, в котором была запущена наша программа (она наследует его от программы-родителя), может повлиять на поведение некоторых функций ввода данных. В одной из предыдущих статей мы уже упоминали о каноническом и неканоническом режиме работы терминала. В каноническом режиме терминал буферизует вводимые данные и передает их программе только после того, как пользователь нажмет Enter. В неканоническом режиме вводимые символы передаются программе немедленно. Режим работы терминала можно изменить с помощью функций cbreak() и nocbreak(). В результате

вызова cbreak() терминал переходит в режим, в котором введенные символы предаются программе, не дожидаясь нажатия Enter, а клавиша BackSpace игнорируется. Терминал выводится из режима cbreak() с помощью вызова функции nocbreak().

Работая с программой *cursedinput*, вы, конечно, заметили, что функция wgetnstr() допускает редактирование вводимой строки с помощью клавиши BackSpace. Поведение этой функции не зависит от режима cbreak()/nocbreak(), но поведение других функций, в частности, getch(), зависит. В режиме nocbreak() функция getch() возвращает управление программе только после того, как пользователь нажмет Enter. Все наши программы (как и большинство программ *ncurses*) устанавливают режим cbreak().

Функция для ввода пароля

Чтобы лучше изучить возможности ввода текста в библиотеке *ncurses*, напишем функцию, предназначенную для ввода пароля. Вместо вводимых символов наша функция будет печатать на экране звездочки (**Рис. 2**). Строку можно будет редактировать с помощью клавиши BackSpace. Мы перепишем программу *cursedinput* так, чтобы вместо своего имени пользователь вводил пароль, который затем сверяется с константой, заданной в программе, и в зависимости от того, совпадут они или нет, будет выводиться сообщение о предоставлении доступа или отказе в нем. Исходный текст программы вы найдете в файле **cursedpassword.c**. Мы приводим только фрагмент, изменившийся по сравнению с программой *cursedinput*.

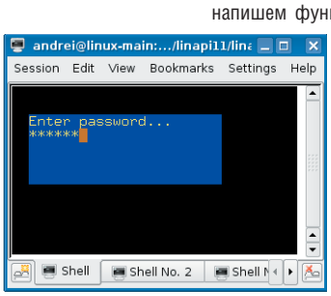


Рис. 2 Ввод пароля с помощью функции *get_password()*.

```
keypad(wnd, TRUE);
wprintw(wnd, “Enter password...\n”);
get_password(wnd, password, MAX_LEN);
wattron(wnd, A_BLINK);
if (strcmp(password, RIGHT_PASSWORD) == 0)
wprintw(wnd, “ACCESS GRANTED!”);
else
wprintw(wnd, “ACCESS DENIED!”);
```

Определенная нами функция get_password() принимает три параметра – идентификатор окна, в котором выполняется ввод, адрес буфера, в который записываются вводимые символы и число, указывающее длину буфера (вместе с завершающим нулем). Прототип функции strcmp(), которую мы используем для сравнения переданной пользователем строки и пароля, находится в файле **string.h**. Рассмотрим теперь саму функцию get_password():

```
void get_password(WINDOW * win, char * password, int max_len)
{
    int i = 0;
    int ch;
    while (((ch = wgetch(win)) != 10) && (i < max_len-1)) {
      if (ch == KEY_BACKSPACE) {
        int x, y;
        if (i==0) continue;
        getyx(win, y, x);
        mvwaddch(win, y, x-1, ‘ ‘);
        wrefresh(win);
        wmove(win, y, x-1);
        i--;
        continue;
      }
      password[i++] = ch;
      wechochar(win, ‘*’);
    }
    password[i] = 0;
    wechochar(win, ‘\n’);
}
```

Функция считывает символы из входного потока с помощью функции wgetch() до тех пор, пока пользователь не нажмет Enter, или пока длина введенной строки не сравняется с максимально допустимой. Для вывода отдельных символов в окно можно применить функцию waddch(), однако мы используем функцию wechochar(), которая эквивалентна вызову waddch() с последующим вызовом wrefresh(). Самая сложная часть функции get_password() связана с обработкой нажатия клавиши BackSpace. Прежде всего, необходимо получить код этой специальной клавиши. По умолчанию при нажатии на специальные клавиши, такие как стрелки, клавиши F1-F12 или BackSpace, терминал генерирует последовательность кодов, представляющих так называемую Esc-последовательность клавиши. Для того чтобы заменить Esc-последовательность одним специальным кодом, необходимо вызвать функцию keypad() с ненулевым вторым параметром (что мы и делаем в главной функции программы). Первым параметром keypad() должен быть идентификатор окна (в нашем случае – wnd).

Вызов keypad() с ненулевым вторым параметром приводит к тому, что клавиши F1-F12 генерируют коды KEY_F1-KEY_F12, клавиши со стрелками – коды KEY_UP, KEY_DOWN, KEY_LEFT, KEY_RIGHT, а клавиша BackSpace – код KEY_BACKSPACE. Описание других кодов специальных клавиш и событий вы найдете на странице man функции getch(). Получив в потоке ввода код KEY_BACKSPACE, мы должны выполнить несколько операций, прежде всего – стереть только что напечатанную звездочку. Для этого нужно получить текущие координаты курсора, сдвинуть его на одну позицию влево и напечатать пробел. Затем курсор снова нужно сдвинуть на одну позицию влево. Получить текущие координаты курсора в окне можно с помощью макроса getyx(). Его первым параметром является идентификатор окна, вторым – переменная, в которой макрос сохранит значение строки курсора, третьим – переменная, в которой будет сохранен столбец курсора. Именно потому, что getyx() – макрос, мы передаем для получения значений переменные, а не указатели на них. Функция mvwaddch() сочетает перемещение курсора и вывод символа. Первый параметр функции – идентификатор окна. За ним следуют новые координаты курсора – строка и столбец. Последним параметром функции является символ, который нужно напечатать. После того, как мы привели в порядок экран, мы уменьшаем на единицу счетчик введенных символов (переменная i). Если переменная i равна нулю, никаких действий не выполняется. Наша функция get_password() будет работать правильно только в режиме cbreak(). Следует отметить, что функция не будет корректно работать с клавишей BackSpace, если при вводе пароля произошел перенос строки.

Окна и мыши

Еще одной полезной возможностью, которую ncurses предоставляет программистам, является поддержка мыши в окне терминала. Рассмотрим программу *cursedmouse* (на диске – файл **cursedmouse.c**), которая регистрирует щелчки левой кнопкой мыши, сделанные пользователем в окне терминала, и распечатывает координаты курсора мыши в момент щелчка. Ради простоты мы не создаем в этой программе никаких окон (кроме окна stdscr, которое создается автоматически).

```
#include <termios.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <stdlib.h>
#include <curses.h>

void sig_winch(int signo)
{
    struct winsize size;
    ioctl(fileno(stdout), TIOCGWINSZ, (char *) &size);
    resizeterm(size.ws_row, size.ws_col);
    nodelay(stdscr, 1);
    while (wgetch(stdscr) != ERR);
    nodelay(stdscr, 0);
}

int main(int argc, char ** argv)
{
```

```
    initscr();
    signal(SIGWINCH, sig_winch);
    keypad(stdscr, 1);
    mousemask(BUTTON1_CLICKED, NULL);
    move(2,2);
    printw("Press the left mouse button to test mouse\n");
    printw("Press any key to quit...\n");
    refresh();
    while (wgetch(stdscr) == KEY_MOUSE) {
      MEVENT event;
      getmouse(&event);
      move(0, 0);
      printw("Mouse button pressed at %i, %i\n", event.x, event.y);
      refresh();
      move(event.y, event.x);
    }
    endwin();
    exit(EXIT_SUCCESS);
}
```

Поддержка мыши в *ncurses* инициализируется с помощью функции mousemask(). Первым параметром этой функции должна быть маска событий мыши, которые следует обрабатывать в программе, вторым параметром может быть указатель на переменную, в которой функция сохранит прежнюю маску событий, или NULL, если прежняя маска нам не нужна. Каждому событию мыши в ncurses соответствует своя константа. Если мы хотим обрабатывать несколько событий мыши, при вызове функции mousemask() мы должны объединить соответствующие константы операцией «ИЛИ» (|). Повторный вызов mousemask() приведет к установке новой маски событий (вызов mousemask() с первым аргументом, равным 0, отключает поддержку мыши).

Рассмотрим некоторые константы, определяющие события мыши. Константа BUTTON1_CLICKED соответствует щелчку левой кнопкой мыши (точнее говоря – щелчку первой кнопкой; будет ли первая кнопка левой кнопкой мыши, зависит от настроек). Константа BUTTON2_PRESSED указывает, что программа должна реагировать на нажатие пользователем второй (обычно – правой) кнопки мыши. Константа REPORT_MOUSE_POSITION указывает, что мы хотим отслеживать движение указателя мыши, а константа ALL_MOUSE_EVENTS заставляет программу реагировать на все события мыши. Более полное описание констант событий вы найдете на man-странице функции mousemask(3x). В качестве результирующего значения функция mousemask() возвращает маску из выбранных нами событий, которые фактически могут быть обработаны. Если функция возвращает 0, значит, работа с мышью в консоли не поддерживается.

Каждый раз, когда в системе происходит одно из «наблюдаемых» событий мыши, в потоке ввода программы появляется специальный символ KEY_MOUSE. Точнее говоря, по умолчанию, в потоке ввода программы Linux появляется Esc-последовательность, соответствующая этому символу, так что в программе *cursedmouse* мы тоже должны вызвать функцию keypad() с ненулевым вторым параметром.

После того, как мы считали из потока ввода специальный символ KEY_MOUSE, мы можем получить более подробную информацию о вызвавшем его событии мыши. Делается это с помощью функции getmouse(). Аргументом функции getmouse() должен быть указатель на структуру MEVENT. Определение структуры MEVENT выглядит следующим образом:

```
typedef struct {
short id;     /* идентификатор для различения нескольких устройств */
int x, y, z;      /* координаты указателя в момент события */
mmask_t bstate;   /* маска событий */
} MEVENT;
```

Координаты указателя возвращаются в формате строка (y), столбец (x). Поле bstate содержит один-единственный бит, соответствующий константе события.

В программе *cursedmouse* мы считываем поступающие во входной поток символы в цикле. Если во входном потоке появляется символ KEY_MOUSE, мы, с помощью функции getmouse(), определяем координаты указателя мыши в момент возникновения события и распечатываем их в левом верхнем углу экрана, а затем переводим курсор туда, куда указывала мышь в момент возникновения события. Появление в потоке ввода символа, отличного от KEY_MOUSE, приводит к завершению программы.

Осталось обратить внимание читателя на обработку сигнала SIGWINCH в программе *cursedmouse*. Изменение размеров экрана при включенной поддержке мыши приведет к появлению в потоке ввода символов Esc-последовательности специального символа KEY_RESIZE (это еще один способ предупредить программу о том, что размеры экрана изменились). В программе *cursedmouse* появление в потоке ввода каких-либо кодов, отличных от KEY_MOUSE, приводит к завершению программы. Для того чтобы избежать этого, в обработчике сигнала SIGWINCH мы опустошаем поток ввода с помощью функции flushinp(). Естественно, этот способ спасения программы от досрочного завершения годится далеко не всегда, ведь в момент изменения размеров окна терминала поток ввода может содержать важную информацию. Все это лишний раз демонстрирует, насколько нетривиальной является обработка изменения размеров экрана в программах *ncurses*.

На этом я заканчиваю (честное слово!) серию статей, посвященную Unix API. Я благодарю вас за внимание, проявленное к этой серии, и надеюсь, что с помощью моих статей вы получили некоторое общее представление о низкоуровневом программировании в Linux/Unix, а самое главное, смогли ответить на вопрос – нужно ли вам все это. LXF

MVC в J2EE

ЧАСТЬ 4 Название статьи выглядит как китайская грамота? Не волнуйтесь – **Антон Черноусов**, вооружившийся книгой Гамма, проведет экспресс-курс по паттернам объектно-ориентированного проектирования.

Наш эксперт

Черноусов Антон
golodnyj@gmail.com.

В предыдущей статье мы говорили о сессионных и контекстных объектах, о том, как создавать и использовать фильтры, и рассмотрели простой пример авторизации пользователя. Сегодня мы рассмотрим паттерн *MVC* и его вариацию для создания web-приложений – *Model2*.

Волшебное слово «паттерн»

Паттерн Model-View-Controller (*MVC*) появился очень давно и достаточно активно использовался в Smalltalk-80. Он предназначен для построения интерфейсов пользователя [1].

Прежде чем окунуться в дальнейшее изучение Java, давайте более подробно остановимся на слове «паттерн». Паттерн – это стандартный способ эффективного решения той или иной задачи в некотором контексте, но для разработчиков программного обеспечения это слово приобрело особое значение. К программному обеспечению этот термин был впервые применен Кристофером Александром [2], который разработал структуру для документирования паттернов и коллекцию паттернов, которые были названы «языками паттернов».

При описании любого паттерна обычно используется следующая структура:

- название и классификация паттерна;
- назначение;
- мотивация для использования;
- рекомендации по применению паттерна;
- структура паттерна;
- участники или элементы паттерна;
- отношения между элементами;
- результаты применения;
- описание идеи реализации паттерна;
- пример кода (по возможности);
- родственные паттерны.

Представленная выше структура описания паттерна применена в книге Э. Гамма, ставшей классическим справочником по паттернам [3].

Фактически, паттерн является средством описания общих решений распространенных задач, то есть это – абстрактное решение технических задач, с которыми приходится сталкиваться программистам [4].

Если сказать еще более простым языком, то паттерн – это стандартное решение стандартной проблемы. Моему коллеге очень нравится следующий пример. *Задача*: построить кирпичную стену, через которую можно видеть; *решение*: строить эту стену с окном. В связи с тем, что количество стандартных задач очень велико, паттернов тоже существует достаточно много.

Что такое MVC?

MVC – паттерн, представляющий объектно-ориентированный метод для разделения логики представления, бизнес-логики и модели данных. *MVC* – не просто удачное решение, а общепринятый образец для построения современных приложений. Иногда паттерн *MVC* называют «архитектурой». На **рис. 1** представлена схема взаимодействия компонентов *MVC*.

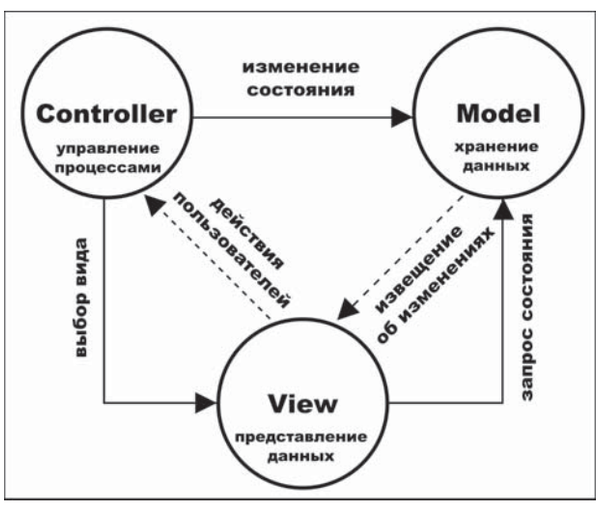


Рис. 1. Архитектура *MVC*.

Месяц назад Мы научились поддерживать сессии и использовать фильтры в наших web-приложениях.

Как вы можете видеть на этом рисунке, основными компонентами являются Модель (Model), Вид (View) и Контроллер (Controller). Компонент Модель представляет собой совокупную модель данных предметной области, реализованной в приложении, например, объект, содержащий в себе информацию о счете пользователя. Вид – это компонент, отвечающий за отображение Модели и реализацию пользовательского интерфейса. Вся логика по обработке действий пользователя сосредоточена в Контроллере.

Иными словами, все данные предметной области, участвующие в работе приложения, сгруппированы в классы, которые предназначены для хранения данных в процессе обработки – эти классы названы Моделью. Классы, реализующие пользовательский интерфейс (какой бы вид они ни принимали: командная строка, GUI или JSP) именуются Видом, а те классы, которые на основании внутренней логики и реакции пользователя принимают решение о выполнении тех или иных действий, принято называть Контроллером.

Зачем нужно использовать паттерн *MVC*? Основным мотивом является облегчение модификации или настройки каждой части в отдельности. Этот паттерн очень полезен в тех случаях, когда нужно создавать компоненты, которые одновременно должны соответствовать требованиям гибкости и удобства сопровождения.

Что не так с гибкостью в приложениях, построенных по другим схемам? В книге Брюса Тейта [5] уделено достаточно много внимания этой проблеме. Вся загвоздка в том, что достаточно часто модель данных и ее представление пользователю сильно переплетены между собой, что вызывает множество проблем при отладке программ и еще больше – при их модификации. Но немаловажной проблемой является и то, что логика работы оказывается рассредоточенной, что опять же не сказывается на приложении положительным образом.

Реализация MVC в J2EE-приложении.

Паттерн *MVC* был предложен для классических настольных приложений, но он практически без изменений может быть использован и для построения приложений J2EE, хотя здесь имеются свои нюансы. Существует две реализации паттерна *MVC* для J2EE-приложений – это *Model1* и *Model2*. В качестве Вида в каждом из них используются JSP, в качестве Модели – JavaBean. Основное отличие подходов заключается в том, каким образом реализован Контроллер: в *Model1* – это JSP, а в *Model2* – сервлет.

Первая реализация основана на логике, которая хранится в страницах. Браузер пользователя поочередно посещает ряд страниц для выполнения какого-либо бизнес-процесса. Такая реализация имеет следующие недостатки.

1. Сложно обеспечить разделение труда между дизайнерами и программистами.
2. Архитектуру *Model1* сложно поддерживать, и она не является гибкой, что особенно критично для больших проектов [6].

Исходя из этих предпосылок, давайте более подробно рассмотрим реализацию компонентов *Model2*, которая чаще всего лежит в основе web-приложений.

Модель – JavaBeans

Прежде всего, Модель в J2EE-приложениях, как и обычных приложениях, удобно реализовывать в виде совокупности JavaBeans. JavaBean'ом может называться любой класс Java, который удовлетворяет достаточно простым требованиям:

- все атрибуты класса должны быть защищенными (protected);
- для каждого атрибута, предназначенного для чтения, должен быть реализован метод вида: PropertyClass getPropertyName(){...};
- для каждого атрибута, предоставляющего возможность записи должен быть реализован метод вида: setPropertyName(PropertyClass propertyName){...};
- обязательно должен быть реализован конструктор без параметров.

Для примера давайте разработаем небольшое web-приложение, отображающее новости. Пусть Моделью приложения будут JavaBean-объекты, инкапсулирующие в себе сведения о новостях. Объект News, содержащий в себе одну-единственную новость, является простым объектом JavaBean и имет только два метода (чтение и запись) для всех своих атрибутов:

```
public class News {
   protected String caption;
   protected String message;
   protected Date date;
   public String getCaption() {
      return caption;
   }
}
```

Второй объект, логически завершающий нашу модель данных, – это NewsTape, который помимо обязательных методов, необходимых JavaBean, имеет дополнительные методы, расширяющие его функциональность, что не запрещено:

```
public void addNews(News news) {
      if (news != null) {
         AllNews.add(news);
      }
   }
```

Связь Вида и Контроллера

Для реализации Вида нам потребуется создать две JSP, одну – для отображения новостей (**view.jsp**), другую – для их добавления (**edit.jsp**). Вторая процедура содержит небольшую форму:

```
<form action="<%=request.getContextPath()%>/
   <%=request.getAttribute("action")%>" method="post">
         <input type="hidden" name="edited"
            value="<%=request.getAttribute("edit.number")%>"/>
      <table>
            <tr><td>Заговок: </td><td><input type="text" name="caption"
 value="<%=request.getAttribute("edit.caption")%>"/></td></tr>
            <tr><td>Сообщение: </td><td><input type="text"
 name="message"
 value="<%=request.getAttribute("edit.message")%>"/></td></tr>
            <tr><td colspan="2" align="center"><input type="submit"
 name="Отправить"/></td></tr>
      </table>
</form>
```

Для реализации контроллера нам необходимо создать сервлет, который в соответствии с запросом пользователя производит выбор необходимого Вида, то есть JSP, например, при помощи следующего метода:

```
   private void handle(HttpServletRequest aRequest, HttpServletResponse
aResponse)
 throws ServletException, IOException {
      aRequest.setCharacterEncoding("utf-8");
      String target =
aRequest.getRequestURI().substring(aRequest.getContextPath().length());
      if (target.equals("/")) {
         outputPage("index.jsp", aRequest, aResponse);
      } else if ("/add".equals(target)) {
         handleAdd(aRequest, aResponse);
      } else if ("/view".equals(target)) {
         handleView(aRequest, aResponse);
      }
   }
```

Вы скажете: «Эй! Да мы подобное уже делали!» И я вам отвечу: «Правильно, вы уже реализовывали контроллер на основе сервлета». Контроллер определяет поведение приложения, то есть интерпретирует действия пользователя и обеспечивает изменение состояния модели или выбор другого представления, или то и другое одновременно.

Из-за особенностей работы web-приложений, выбор Вида из Контроллера можно произвести двумя способами: вызвав метод forward у экземпляра класса RequestDispatcher или вызвав метод sendRedirect у экземпляра класса HttpServletResponse. В то же время,

воздействия пользователей на «Контроллер», в основном, могут быть переданы посредствам запросов, то есть нажатием на ссылки, или в качестве результатов работы форм. Управление работой приложения происходит благодаря методу handle нашего сервлета. Такой вид организации обработки принято назвать «реализация стратегии **Servlet Front** паттерна *Front Controller*». Хотя, если быть более точным, реализован подвид стратегии **Servlet Front – Dispatcher in Controller**.

Видимо, такое обилие терминов сбивает с толку, поэтому давайте вкратце рассмотрим стратегии реализации паттерна *Front Controller*, предназначенные для реализации «Контроллера» в паттерне *MVC*.

1. Стратегия **Servlet Front** фактически соответствует поведению контроллера в архитектуре *Model2*, описанной выше. «Контроллер» редко выполняет все функции, но в случае, если это происходит, и функциональность, отвечающая за перенаправление пользователя к другим «Видам», полностью реализована в нем, то такая реализация называется **Dispatcher in Controller**. Еще один подвид данной стратегии – **Base Front**, она отличается тем, что реализован не один контроллер, а несколько, причем все они расширяют некий базовый контроллер, в котором реализована базовая функциональность. Применение этой стратегии не всегда оправдано.
2. Стратегия **JSP Front** фактически соответствует поведению контроллера в архитектуре *Model1*, описанной выше, и мы не будем на ней останавливаться, потому как она практически нигде не применяется.
3. **Mapping Controller** – это стратегия, которая тем или иным образом применяется при построении практически любого web-приложения. Технология реализована через дескриптор развертывания приложения и имеет следующие подвиды: **Physical Resource Mapping** – когда ресурс отображается по месту его физического расположения в каталогах, **Logical Resource Mapping** – когда ресурс отображается в соответствии с некоторой логикой группирования функции приложения, и **Multiplexed Resource Mapping** – это смешивание обоих методов размещения ресурсов.
4. Стратегия **Filter Controller** – это реализация контроллера в виде фильтра, который перехватывает все запросы пользователя и в соответствии с внутренней логикой обеспечивает тот или иной бизнеспроцесс.
5. **Command and Controller** – это стратегия, основанная на применении двух замечательных паттернов, *Command* и *Factory Method*, и, на мой взгляд, одна из самых удачных стратегий. Ее применение мы обсудим в одной из наших следующих статей.

Извещение об изменениях

Рис. 2. Схема взаимодействия некоторых объектов электронного магазина.

Последний вопрос, который необходимо осветить для полного обзора паттерна *MVC* – это извещение Вида об изменениях состояния Модели. К сожалению, в большинстве случаев в J2EE нельзя выполнить оповещение «Вида» за исключением случаев, когда приложение построено с применением ряда современных технологий. Примером такой кооперации может служить AJAX.

И все же, давайте рассмотрим механизм оповещения одного объекта об изменениях, произошедших в другом объекте. Реализация механизма оповещения необходима в следующем случае: существует как минимум один объект, рассылающий уведомления, и имеется не менее одного получателя сообщений, причем количество сообщений и их состав может меняться во время работы, а также различаться у разных элементов приложения.

Классическим примером такого поведения может служить электронный магазин, где необходимо обеспечить взаимодействие корзин покупателей, их счетов и склада с товаром. На **рис. 2** представлена схема взаимодействия рассмотренных объектов. Пользователь выбирает товар, и он помещается в корзину, корзина уведомляет объект – счет, где формируется стоимость товара и доставки, и склад, где происходит бронирование или подготовка к дополнительному заказу товара у поставщиков.

В основе описанного процесса лежит реализация паттерна *Observer*. Его цель – это организовать зависимость «один ко многим» между объектами так, чтобы при изменении состояния одного объекта автоматически уведомлялись и обновлялись все его зависимые объекты.

Считается, что *Observer* является одним из главных паттернов объектно-ориентированного программирования, по крайней мере, он широко применяется в настольных приложениях. Особенно хорошо его влияние видно в приложениях Swing. Для применения этого шаблона необходимо ответить на следующие вопросы:

1. Кто является объектом наблюдения, а кто наблюдателем?
2. Когда объект наблюдения должен послать уведомление своим наблюдателям?
3. Что должен делать наблюдатель при получении уведомления?
4. Как должно начинаться и как заканчиваться взаимодействие для организации наблюдения?

Естественно, что на все эти вопросы уже найдены ответы, поэтому давайте рассмотрим **рис. 3**, на котором изображена диаграмма классов паттерна *Observer*.

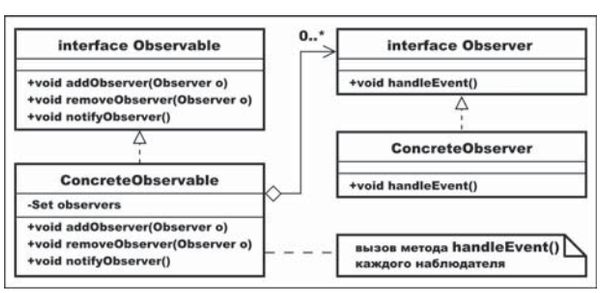


Рис. 3. Диаграмма классов паттерна *Observer*.

В диаграмме определены два интерфейса: обозреватель (Observer) и объект наблюдения (Observable). Класс, реализующий интерфейс Observable, предоставляет методы для подключения и отключения обозревателей, содержит в себе текущий список наблюдателей (атрибут observers), а также имеет метод оповещения всех наблюдателей – notifyObserver(). Итак, используем интерфейсы для создания небольшого примера:

```
public interface Observer {
  public void handleEvent();
}
public interface Observable {
  public void addObserver(Observer o);
```

```
    public void removeObserver(Observer o);
    public void notyfyObserver();
}
```

Создадим объект наблюдения – пусть это будет совсем простенькая модель погоды, которая содержит в себе сведения о температуре окружающей среды (атрибут temperature):

```
public class Weather implements Observable {
    private Set<Observer> observers = new HashSet();
    private int temperature;

    public void addObserver(Observer o) { observers.add(o); }
    public void removeObserver(Observer o) { observers.remove(o); }
    public void notyfyObserver() {
        for (Observer o : observers) {
            o.handleEvent();
        }
    }

    public Weather() { this.temperature = 0;}
    public int getTemperature() { return temperature; }
    public void setTemperature(int temperature) {
        this.temperature = temperature;
        notyfyObserver();
    }
}
```

Теперь создадим класс для ведения наблюдения за температурой, то есть термометр:

```
public class Thermometer implements Observer {
    private int temperature;
    private Weather weather;

    public Thermometer(Weather weather) {
        this.weather = weather; this.handleEvent();
    }
    public int getTemperature() { return temperature; }
    public void setWeather(Weather weather) {
        this.weather = weather; this.handleEvent();
    }
    public void handleEvent() {
        if (weather != null) {
            temperature = weather.getTemperature();
            System.out.println("температура : " +  temperature);
        }
    }
}
```

Чтобы проверить работоспособность оповещения, попробуйте выполнить следующий код:

```
        Weather currentWeather = new Weather();
        Thermometer theThermometer1 = new Thermometer(currentWeather
);
        Thermometer theThermometer2 = new Thermometer(currentWeather
);
        currentWeather.addObserver(theThermometer1);
        currentWeather.addObserver(theThermometer2);
        System.out.println("Изменение температуры 1");
        currentWeather.setTemperature(10);
        System.out.println("Изменение температуры 2");
        currentWeather.setTemperature(15);
```

Вы убедитесь в том, что у каждого наблюдателя вызывается метод handleEvent(), что влечет за собой изменение его внутреннего состояния.

Подведем итоги. В этой статье мы рассмотрели реализацию шаблона *MVC* в J2EE и ряда других паттернов, которые позволяют реализовывать масштабируемые приложения и разобрались в требованиях, предъявляемых к классам так называемых JavaBeans. Кроме того, мы вкратце ознакомились со способами организации Контроллера и обратили свое внимание на процесс оповещения объектов на основе паттерна *Observer*. Пожалуй, на сегодня хватит. LXF

Литература

1 Krasner G.E., Pope S.T. *A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80.//* Journal of Object-Oriented Programming. – 1988. – № 3, ч. 1. – С. 26-49.

2 Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I. and Angel S. *A Pattern Language*. – New York: Oxford University Press, 1977. – 1216 с.

3 Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. *Приемы объектно-ориентированного проектирования. Паттерны проектирования*. – СПб.: Питер, 2001. – 368 с.

4 Мак-Карти Д., Мак-Карти М. *Программируем командный дух*. – СПб.: Символ-Плюс, 2004. – 416 с.

5 Тейт Б. *Горький вкус Java: Библиотека программиста*. – СПб.: Питер, 2003. – 333 с.

6 Курняван Б. *Создание Web-приложений на языке Java*. – Москва: Издательство "ЛОРИ", 2005. – 880 с.

Через месяц Мы научимся подключаться к различным базам данных посредством JDBC.

Справочно-поисковый

ЧАСТЬ 9 *TeX* создавался для верстки книг и сегодня **Евгений Балдин** рассмотрит всё то, без чего немыслимо серьезное произведение – оглавление, предметный указатель, сноски и т.д..

На этом же этаже располагалось книгохранилище. По поводу его размеров рассказывали, что в глубине, в полукилометре от входа, идёт вдоль стеллажей неплохое шоссе, оснащённое верстовые столбами.

«Понедельник начинается в субботу»
Аркадий и Борис Стругацкие.

Книги делятся на те, что читаются один раз и те, что многократно перечитываются. Наличие информации о структуре книги повышает ценность любого текста. Отсутствие этой информации – прямой намёк, что после прочтения произведение следует забыть и выбросить.

Справочно-поисковый аппарат издания позволяет читателю облегчить и ускорить поиск имеющихся в книге объектов. В качестве элементов, из которых складывается справочного-поисковая система, можно упомянуть рубрикацию, оглавление, колонтитулы, ссылки, подстрочные примечания, алфавитный указатель и библиографию.

Этот аппарат существует исключительно для читателя, и он достаточно трудоёмок при создании, но сложности не должны пугать истинных энтузиастов в деле создания текстов, так как их преодоление значительно повышают ценность серьёзного произведения.

Пример рубрикации

А. В. Тор

Содержание

1. Раздел 1
1.1. Подраздел 1
1.1.1. Подподраздел 1
Заключение 1
А. Приложение 1

1. Раздел

Основной элемент рубрикации.

1.1. Подраздел

Вспомогательный элемент рубрикации.

1.1.1. Что-то более мелкое чем подраздел

Вспомогательный для вспомогательного. В содержании выводится краткая версия заголовка.

Параграф Важный параграф.

Подпараграф Параграф чуть менее важный.

Раздел, отсутствующий в содержании

Всяко бывает. Иногда и такое нужно.

Заключение

Заключение, в отличие от, скажем, раздела 1.1 на странице 1, нумеровать не надо, но в содержании отразить необходимо.

А. Приложение

› **Пример рубрикации и оглавления.**

Рубрикация и оглавление

Нужны ли книге оглавление или содержание? Любой скажет: что за вопрос, конечно, нужны. И не только в книге научной и деловой. В любой.

А.Э. Мильчин. «Культура издания»

Для оформления разделов в основном используются команды секционирования: \section, \subsection, \subsubsection, \paragraph и \subparagraph.

Команды перечислены в порядке убывания значимости при рубрикации. Обычно любой сколько-нибудь сложный текст следует начинать с планирования структуры, то есть создать шаблон, примерно следующего вида:

```
\documentclass[a4paper,12pt]{ncc}
\usepackage[warn]{mathtext}
\usepackage[T2A]{fontenc}
\usepackage[koi8-r]{inputenc}
\usepackage[english,russian]{babel}
\usepackage{indentfirst}
\title{Пример рубрикации}
\author{A.\,B.~Top}
\begin{document}
\maketitle{}
\tableofcontents{}

\section{Раздел}
\label{sec:section}
Основной элемент рубрикации.

\subsection{Подраздел}
\label{sec:subsection}
Вспомогательный элемент рубрикации.

\subsubsection[Подподраздел]{Что-то более мелкое чем подраздел}
\label{sec:subsubsection}
Вспомогательный для вспомогательного. В содержании выводится
краткая версия заголовка.

\paragraph{Параграф}
\label{sec:paragraph}
Важный параграф.
```

Месяц назад Мы научились создавать средствами *LaTeX* эффективные презентации.

аппарат издания

```
\subparagraph{Подпараграф}
\label{sec:subparagraph}
Параграф чуть менее важный.

\section*{Раздел, отсутствующий в содержании}

Всяко бывает. Иногда и такое нужно.

\section*{Заключение}
\label{sec:afterwords}
\addcontentsline{toc}{section}{Заключение}

Заключение, в отличие от, скажем, раздела~\ref{sec:subsection}
на странице~\pageref{sec:subsection}, нумеровать не надо, но в
содержании отразить необходимо.

\appendix

\section{Приложение}
\label{appendix}

\end{document}
```

Результат компиляции кода представлен на иллюстрации. Кроме самих заголовков разделов, созданных с помощью команд секционирования, в начале документа создаётся оглавление. За это отвечает команда \tableofcontents. При каждой компиляции информация о разделах собирается в файле с тем же именем, что и у tex-файла, но с расширением toc. При следующей компиляции обновлённая информация о разделах используется для создания оглавления.

Уровень, до которого информация отображается в оглавлении, можно поменять, изменив значение переменной tocdepth, например, так:

```
\setcounter{tocdepth}{2}
```

В этом случае будет показана информация о разделах вплоть до второго уровня. Раздел типа \section соответствует первому уровню секционирования, \subsection – второму, и так далее.

Кроме оглавления, можно вывести список иллюстраций \listoffigures и таблиц \listoftables. Информация об иллюстрациях и таблицах автоматически собирается в файлах с расширениями **.lof** и **.lot**.

Для добавления какой-то информации в оглавление в обход команд секционирования можно воспользоваться командой

```
\addcontentsline{toc}{«уровень раздела»}{«строка в оглавлении»}
```

У этой команды три аргумента. Первый аргумент соответствует расширению файла (**.toc**, **.lof** или **.lot**), в который добавляется «строка в оглавлении». Уровень раздела определяется именами команд секционирования, то есть section, subsection и так далее. Команды *LaTeX* при передаче в файлы списков следует защищать командой \protect, дабы избежать проблемы с «хрупкими» инструкциями.

Команда \appendix отделяет приложение от основного текста. После её вызова правила нумерация разделов изменяется. \appendix тоже является командой секционирования.

Ссылки и примечания

Иногда не хочется разбивать канву повествования, и в то же время необходимо как-то вставить пояснение. Это можно сделать, просто сославшись на какой-то другой фрагмент текста с помощью ссылки или вставив пояснение на этой же странице с помощью подстрочного примечания.

Механизм ссылок

В примере, демонстрирующем работу команд секционирования, вслед за каждой такой командой ставилась метка с помощью инструкции \label. Метка представляет собой последовательность ASCII-символов. Префикс sec: автоматически добавляется AUCTeX во время вставки метки при редактировании tex-файла в *Emacs*. Этот префикс существует исключительно для удобства автора, так что можно обойтись без него. При компиляции документа информация о имеющихся метках добавляется в файл с расширением **.aux**. Для того, чтобы извлечь эту информацию, то есть номер раздела (команда \ref) или номер страницы (\pageref), необходимо пропустить текст через *LaTeX* ещё раз.

Для того, чтобы можно было сослаться на внешний документ, следует воспользоваться пакетом *xr*. В этом случае в преамбуле необходимо добавить примерно такие инструкции:

```
\usepackage{xr}
\externaldocument[EXT-]{externaldoc}
```

Это позволяет получить доступ к меткам файла **externaldoc.tex**. Обращение к меткам, как и обычно, осуществляется помощью команд \ref/\pageref, только перед именем метки добавляется префикс EXT-. Можно обойтись и без префикса, так как этот параметр является опциональным, но в этом случае повышается вероятность конфликта из-за существования одинаковых меток.

Ссылаться можно не только на разделы. Метки внутри нумерованных окружений, типа equation (выключенные математические формулы[1]) или theorem (теоремы), принимают их номер. Это так же касается рисунков (окружение figure) и таблиц (окружение tabular). В этом случае \label должна следовать сразу за командой \caption, формирующей подпись к плавающему объекту.

Общего рецепта создания ссылок на электронные ресурсы нет. Проще всего использовать команду \url из одноимённого пакета:

`Моя WWW-страничка находится по адресу \url{http://www.inp.nsk.su/~baldin/}.`	Моя WWW-страничка находится по адресу http://www.inp.nsk.su/~baldin/.

В адресной строке должны отсутствовать символы %, #, ^, и она не должна заканчиваться символом \. Если есть желание уйти и от этих ограничений, то аналогично команде \verb инструкцию можно использовать и так: \url!http://www.адрес.ru!.

В плане создания и управления гиперссылками также интересен пакет *hyperref*, который предоставляет схожую функциональность и позволяет создавать гиперссылки в pdf-документах, но это уже совсем другая история:

```
%загрузка пакет hyperref
\usepackage[unicode=true]{hyperref}
```

Подстрочные примечания

Подстрочное примечание формируется с помощью команды \footnote. Правила оформления примечаний прописываются в определении

[1] Для правильной ссылки на номера формул вместо команды \ref следует использовать инструкцию \eqref.

» класса, и без особых на то причин менять их не стоит. Примечание, если позволяет место, печатается на той же странице, где помещена ссылка, и отделяется от текста разделительной линией.

Примечание можно добавлять и внутри окружения minipage, но тогда оно печатается внутри окружения:

```
\begin{minipage}{1.0\linewidth}
    Ссылки раз\footnote{Сноска.} и
  два\footnote[26]{Подстрочное примечание.}.
\end{minipage}
```

Ссылки раз[a] и два[z].

[a]Сноска.
[z]Подстрочное примечание.

Необязательный параметр \footnote позволяет присвоить примечанию значение по выбору пользователя.

Для того чтобы можно было сделать сноску внутри заголовка раздела, необходимо защитить инструкцию \footnote командой \protect – команда создания подстрочного примечания является «хрупкой»:

```
\section{Заголовок\protect\footnote{Подстрочное примечание.}}
```

Иногда, в сложных ситуациях, например, когда требуется сделать подстрочное примечание внутри бокса, для формирования сноски требуется прибегнуть к независимым командам создания ссылки и создания примечания:

```
\footnotemark[num]
\footnotetext[num]{«сноска»}
```

Необязательный параметр num, как и в случае \footnote, позволяет формировать свою нумерацию. Для хранения текущего номера ссылки используется счётчик footnote.

Колонтитулы

Правила формирования колонтитулов целиком зависит от выбранного класса документа. Если же хочется изменить значения по умолчанию, то проще всего выбрать стиль страницы **myheadings** и сформировать колонтитулы:

```
\pagestyle{myheadings}
\markboth{«левый колонтитул»}{«правый колонтитул»}
```

Если печать односторонняя, то достаточно воспользоваться командой \markright, которая имеет только один аргумент.

Для полного управления содержимым колонтитулов лучше всего подходит пакет *fancyhdr*. Подробно об этом пакеты было рассказано ранее в статье «Вёрстка», раздел «Стили страницы» (LXF89).

Библиография

Книги создаются не в безвоздушном или бесknижном пространстве.
А.Э. Мильчин. «Культура издания»

Хорошая книга представляет из себя ценность, но даже самая лучшая книга не в состоянии охватить абсолютно все аспекты рассматриваемой в ней темы. Книги существуют в книжном пространстве. Всегда можно найти что-то, на чём книга основывалась; что-то, что развивает основную идею; и что-то, что позволяет взглянуть на главную тему с другой стороны. Список литературы только украшает книгу.

Для ссылок на литературу используется команда

```
\cite[«комментарий»]{«список меток»}
```

Метки либо формируются автором самостоятельно внутри окружения thebibliography:

```
Полезно почитать книгу \cite[Роженко]{rozenko-2005}.

\begin{thebibliography}{9}
\bibitem{rozenko-2005}Роженко А.И. Искусство вёрстки в
\LaTeX'е. \newblock --- Новосибирск: Изд. ИВМиМГ СО~РАН,
2005. 398~с.
\end{thebibliography}
```

либо используется механизм *BibTeX*. Команда **\newblock** позволяет логически разделить разные по смыслу элементы. В качестве обязательного аргумента окружения thebibliography требуется передать текст, соответствующий самой широкой метке, для выравнивания. То есть, если список литературы содержит меньше 10 записей, то достаточно передать однобуквенную фразу, например, «9», а в случае двузначного числа книг в списке уже потребуется «99», и так далее.

Полезно почитать книгу [1, Роженко].

Список литературы

[1] Роженко А.И. Искусство вёрстки в ИТЕХ'е. — Новосибирск: Изд. ИВМиМГ СО РАН, 2005. 398 с.

› Пример библиографической ссылки.

Как и в случае с перекрёстными ссылками, информация о списке литературы заносится в файл с расширением aux, то есть для правильного его отображения необходимо два прохода *LaTeX*.

Для того чтобы можно было использовать кириллицу в метках для цитирования литературы, то есть иметь возможность написать что-то вроде:

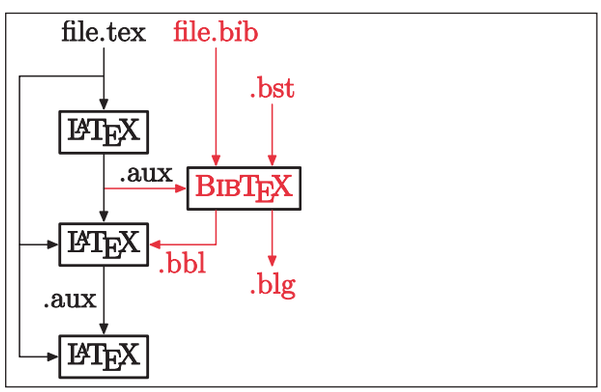
```
\cite{Котельников-2004}
...
\bibitem{Котельников-2004}
```

Следует воспользоваться пакетом *citehack*:

```
\usepackage{citehack}
```

Из названия пакета очевидно, что это «хак» со всеми вытекающими последствиями и что им не следует злоупотреблять.

BibTeX



› Конвейер LaTeX+BibTeX. tex – LaTeX-исходник, bib – библиографическая база, bst – стилевой файл для библиографии, blg – log-файл BibTeX, bbl – отсортированный список литературы, aux – информация о ссылках.

Список литературы можно оформлять вручную. Есть какой-никакой стандарт, например, тот же ГОСТ 7.80-00[2] или ГОСТ 7.1-84. Его можно просто взять и следовать подробной инструкции. Но далеко не все издательства подчиняются этому стандарту, в котором, например, нет информации о том, как нужно оформлять www-ссылки. Да и вообще, список сопутствующей литературы – это нечто большее, чем просто довесок к статье или книге: это вполне самостоятельный фрагмент информации, который очень полезно уметь представлять по-разному.

Для решения этой проблемы Орен Поташник разработал программу *BibTeX*, которая сама формирует окружение thebibliography, получая

[2] ГОСТ на оформление библиографического указателя, принятый в 2000 году. Правила оформления могут нравиться или не нравиться, но это всё-таки хоть какой-то стандарт.

информацию из текстовой библиографической базы. Структура библиографической базы *BibTeX* является довольно распространённым форматом, который использует в том числе и Google Scholar (**http://scholar.google.com/**), не говоря уж о том, что основной архив электронных препринтов **http://xxx.lanl.gov** предоставляет библиографическую информацию исключительно в виде записей *BibTeX*.

Из программного обеспечения, позволяющего работать с *BibTeX*, следует упомянуть встроенный в *Emacs* пакет *RefTeX* и *JabRef* **http://jabref.sourceforge.net/**. Тот, кто не освоил *Emacs*, и кому не нравится Java, может поискать программные пакеты *gBib* и *KBib* для Gnome и KDE соответственно. Простой конвертер *BibTeX2html* позволяет получить список литературы в html-виде. Естественно, и простое редактирование текстового файла вручную также никто не отменял.

Как правило, библиографическая база в формате *BibTeX* хранится в файле с расширением **.bib**. Перед тем, как с помощью команды \bibliography подключить базу к *LaTeX*-исходнику, нужно выбрать стиль сортировки библиографии:

```
\bibliographystyle{«стиль»}
\bibliography{«имя bib-файла»}
```

В *LaTeX* есть четыре стандартных стиля для формирования списка литературы:

- **plain** – открытый стиль. Библиографические записи помечаются порядковыми номерами и сортируются в алфавитном порядке. Чтобы правильно отсортировать библиографию на русском языке, необходимо вместо *BibTeX* воспользоваться командой *BibTeX8*, указав с помощью ключа --csfile соответствующее правило сортировки[3].
- **unsrt** – несортирующий стиль. В отличии от **plain**, порядок представления списка литературы определяется порядком цитирования библиографии в тексте.
- **alpha** – алфавитный стиль. Вместо нумерации библиографии используются имена меток. Литература сортируется по меткам.
- **abbrv** – аббревиатурный стиль. Вместо полных имён авторов, названий месяцев и журналов печатаются сокращения. Сортировка и нумерация соответствует стилю **plain**.

Максим Поляков разработал стили для *BibTeX*, соответствующие ГОСТ 7.80-00 и GOST 7.1-84: **gost780s/gost71s**, аналогичные **plain** и **gost780u/gost71u**, аналогичные **unsrt**. Описание этих стилей представлено в стандартной документации в виде файлов **gost780.pdf** и **gost71.pdf**. Многие журналы, принимающие публикации в *LaTeX*, имеют свои собственные *BibTeX*-стили. В стандартной поставке *TeXLive 2007* идёт более 200 различных библиографических стилей – есть из чего выбирать.

Список литературы

[1] *Гуссенс, М.* Путеводитель по пакету LaTeX и его расширению LaTeX 2ε: Пер. с англ. / М. Гуссенс, Ф. Миттельбах, А. Самарин. — М.: Мир, 1999. — 606 с.

[2] *Львовский, С. М.* Набор и вёрстка в системе LaTeX. / С. М. Львовский. — М.: МЦНМО, 2003. — 448 с. — 3-е изд., испр. и доп.

[3] *Котельников, И. А.* LaTeX по русски. / И. А. Котельников, П. З. Чеботаев. — Новосибирск: Сибирский Хронограф, 2004. — 496 с. — 3-е изд., перераб. и доп.

[4] *Рожeнко, А. И.* Искусство вёрстки в LaTeX'е. / А. И. Рожeнко; Под ред. А. Алексеева. — Новосибирск: Изд. ИВМиМГ СО РАН, 2005. — 398 с.

Список литературы, оформленный с помощью *BibTeX*. Стиль gost780u.

В дополнение к команде цитирования \cite в случае подключения библиографической базы можно использовать инструкцию \nocite. Команда \nocite не создаёт никакой ссылки в тексте, но упомянутая запись отображается в списке литературы.

Сама по себе база состоит из записей вида:

```
@book{Gussens-1999,
 author = {М. Гуссенс and Ф. Миттельбах and А. Самарин},
 title = {Путеводитель по пакету \LaTeX{} и его
          расширению \LaTeXe: Пер. с англ.},
 year = {1999},
 isbn = {5-03-003325-4},
 publisher = {Мир},
 address = {М.},
 numpages = {606},
 language = {russian},
 OPTnote = {}
}
```

После знака «коммерческое at» @ идёт тип записи. В фигурных скобках вслед за меткой через запятую перечисляются пары ключ-значение. *BibTeX* поддерживает определённый набор типов записей, каждому из которых соответствуют свои обязательные и необязательные поля[4]. Если не заполнено обязательное поле, то при компиляции *BibTeX* генерирует ошибку. Имеются следующие стандартные типы записей (*BibTeX* не чувствителен к регистру):

- **Aritcle** – статья в журнале. Обязательные поля: author, title, journal, year. Необязательные поля: volume, number, pages, month, note, annote.
- **Book** – книга. Обязательные поля: author или editor, title, publisher, year. Необязательные поля: volume, number, series, address, edition, month, note, annote.
- **Booklet** – брошюра. Обязательное поле: title. Необязательные поля: author, howpublished, address, month, year, note, annote.
- **Conference** или **InProceedings** – статья, опубликованная в трудах конференции. Обязательные поля: author, title. Необязательные поля: crossref, booktitle, pages, year, editor, volume, number, series, address, month, organisation, publisher, note, annote.
- **Proceedings** – труды конференции. Обязательные поля: title, year. Необязательные поля: booktitle, editor, volume, number, series, address, month, organisation, publisher, note, annote.
- **InBook** – ссылка на часть книги, то есть на её главу, раздел или просто на определённый набор страниц. Обязательные поля: author или editor, title, chapter, publisher, year. Необязательные поля: volume или number, series, type, address, edition, month, pages, note, annote.
- **InCollection** – часть книги со своим заглавием. Обязательные поля: author, title, booktitle. Необязательные поля: crossref, pages, publisher, year, editor, volume или number, series, type, chapter, address, edition, month, note, annote.
- **Manual** – техническая документация. Обязательное поле: title. Необязательные поля: author, organisation, address, edition, month, year, note, annote.
- **PhdThesis** – диссертация. Обязательные поля: author, title, school, year. Необязательные поля: address, month, note, annote.
- **MastersThesis** – дипломная работа. Обязательные и необязательные поля такие же, как у **PhdThesis**.
- **TechReport** – отчёт. Обязательные поля: author, title, institution, year. Необязательные поля: type, numer, address, month, note, annote.
- **Unpublished** – неопубликованный авторский текст. Обязательные поля: author, title, note. Необязательные поля: month, year, annote.
- **Misc** – то, что не подходит для других типов записей. Необязательные поля отсутствуют. Необязательные поля: author, title, howpublished, month, year, note, annote.

Значение полей в большинстве случаев понятно из их названия. Исключением, пожалуй, является поле crossref, в качестве значения которого можно передать ссылку на другую запись, откуда *BibTeX* при трансляции возьмёт значения всех недостающих полей записи. Из особенностей следует упомянуть, что авторы в поле author разделяются с помощью союза and.

Кроме перечисленных стандартных полей при использовании библиографических ГОСТ-стилей **gost780u/gost71u** и **gost780s/gost71s** можно использовать поля numpages – число страниц и language – влияет на сортировку при выборе стиля типа plain. Допустимые значения для поля language – russian, ukrainian и english по умолчанию.

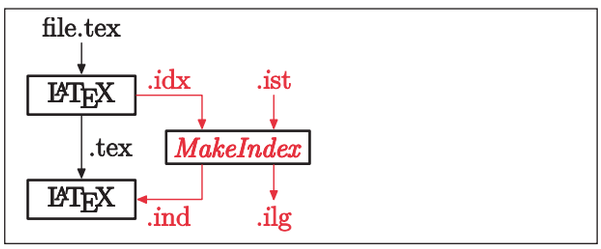
[3] В стандартной поставке *LaTeX* есть правило для сортировки для кодовой страницы **cp866 cp866rus.csf**. На основе этого файла можно создать правило для другой кодовой страницы.

[4] Если к названию необязательного поля добавить OPT (note->OPTnote), то такие такие поля игнорируются, даже если присутствуют в записи.

Алфавитный указатель

Указатель значительно повышает ценность любой книги, если она из тех, что читают не только насквозь, но и выборочно.

А.Э. Мильчин «Культура издания»



› Конвейер LaTeX+MakeIndex. tex – LaTeX-исходник, idx – не отсортированный индекс (полуфабрикат), ist – стилевой файл для указателя, ilg – log-файл MakeIndex, ind – отсортированный указатель.

Для создания «полуфабриката» алфавитного указателя в преамбуле помещается инструкция

```
\makeindex
```

Слова, которые нужно поместить в указатель, отмечаются с помощью команды \index.

```
Указатель\index{Предметный указатель} …
```

Сама по себе команда \index игнорируется, но всё, что в ней отмечается вместе с информацией о положении команды, заносится в файл с расширением **.idx**. Имя idx-файла по умолчанию соответствует имени основного документа. Полуфабрикатом idx-файл является потому, что записи хранятся в нём в неотсортированном виде. С помощью программы сортировки *rumakeindex* получается отформатированный правильным образом указатель в файле с расширением **.ind**. Этот файл уже можно вставить в документ:

```
\input{«ind-файл»}
```

Это же делает и команда \printindex из пакета *makeidx*. В дополнение к подключению индекса команда \printindex проверяет существование индексного файла и не даёт *LaTeX* генерировать ошибку в случае его отсутствия. Пакет *makeidx* содержит ещё несколько полезных команд для создания индекса, поэтому его в любом случае имеет смысл загрузить.

Программа *rumakeindex* является простейшим скриптом, где с помощью *sed* кириллические буквы из внутреннего представления *LaTeX* переводятся в koi8-r и правильным образом сортируются с использованием стандартного механизма *makeindex*. Если необходимо отсортировать индексный файл для включения в текст, использующий другую кодовую страницу, или вас просто не устраивают правила сортировки, этот скрипт легко переделать.

Оригинальная программа сортировки индекса *makeindex* (подробная документация представлена в файле **makeindex.dvi**) была написана довольно давно и, естественно, не учитывала национальных особенностей других языков, кроме английского и немецкого. Впрочем, она оказалась слишком гибка, чтобы полностью отказаться от неё в пользу другого механизма сортировки, уже нормально поддерживающего интернационализацию. Наиболее вероятным претендентом на замену уже долгое время является *xindy* (**http://www.xindy.org/**), который «из коробки» поддерживает множество языков вплоть до клингонского, но до сих пор отсутствует в основных *LaTeX*-дистрибутивах.

Аргумент команды \index может содержать любые символы, кроме !, “, @ и |. Их специальное значение проявляется только внутри команды. Чтобы убрать специальное значение этих символов внутри \index необходимо добавить перед ними символ “ (двойную кавычку). На рисунке представлен пример простейшего указателя. Ниже будут раскрыты методы его создания.

Предметный указатель

MakeIndex, **110**

Указатель, 110–111
makeindex, *см.* rumakeindex
rumakeindex, 110, 111
xindy, 111

› Пример готового алфавитного указателя.

Для формирования многоуровневых иерархических указателей используется разделитель в виде восклицательного знака !:

```
% на страницах 110 и 111
\index{Указатель!rumakeindex}
```

Команды \index с одинаковыми аргументами группируются в одну запись с полным списком страниц.

Символ вертикальной черты | используется для отделения видимого аргумента от управляющих знаков. Команда \see (перекрёстная ссылка на другую запись), определённая в пакете *makeidx*, должна идти в индексе вслед за этим разделителем:

```
\index{Указатель!makeindex|see{rumakeindex}}
```

С помощью вертикальной черты можно также сформировать указатель на диапазон страниц:

```
% на странице 110
\index{Указатель|(}
много текста
% на странице 111
\index{Указатель|)}
```

Иногда номер страницы нужно как-то выделить. Команды выделения текста также должны идти после вертикальной черты:

```
% Выделяем страницу 110 жирным шрифтом
\index{MakeIndex@\textit{MakeIndex}|textbf}
```

Конструкция «ключ»@«запись» используется для правильной сортировки внутри \index. По «ключу» производится сортировка, а «запись» выводится в предметном указателе.

Заключение

Написать любую книгу безумно тяжело. Сделать её полезной – почти невозможно. Справочно-поисковый аппарат издания – лишь инструмент на этом пути, но инструмент, неоднократно проверенный грамотным человечеством. Можно путь пройти и без него, но с ним будет гораздо интереснее. LXF

» **Через месяц** Все (или почти все) о вёрстке таблиц в *LaTeX*.

Новая серия! Обработка видео средствами открытой программы для нелинейного монтажа

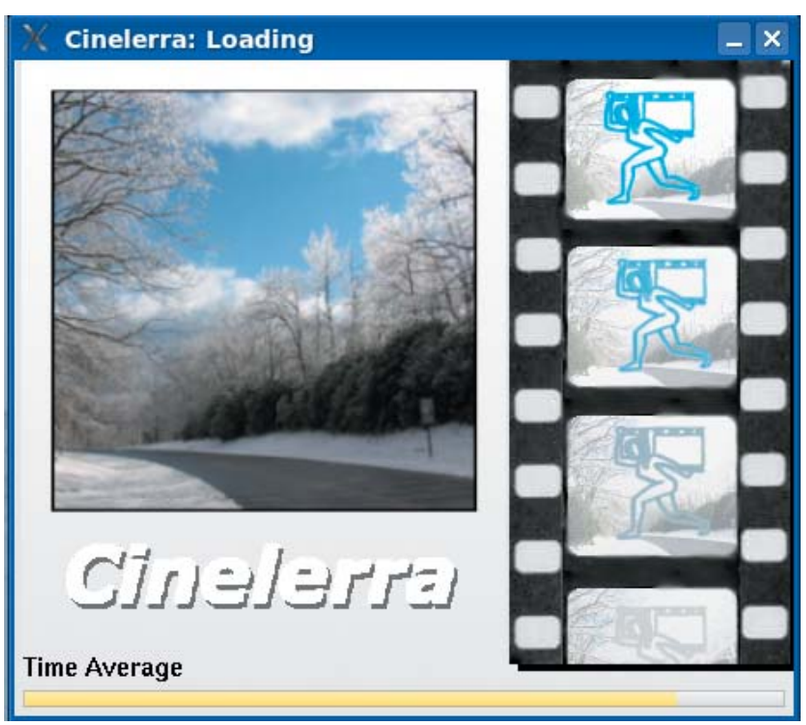
Хроматический ключ

ЧАСТЬ 1 Динозавры заполонили улицы мирного города и жители в страхе разбегаются... **Петр Семилетов** покажет, как создать картину грядущего апокалипсиса средствами *Cinelerra*!

Сегодня мы начинаем серию статей о монтаже видео в программе нелинейного монтажа *Cinelerra* (см. LXF85). Продукт этот, равно как и его установка, не отличается простотой. Нужно отметить, что существует две одновременно разрабатываемые версии – просто *Cinelerra* (оригинальная версия от Heroine Virtual Ltd.) и *Cinelerra CV* (версия, создаваемая сообществом программистов независимо от Heroine Virtual Ltd.). Обе ветки иногда обмениваются исходным кодом. В «исходники» включено много библиотек – это *ffmpeg*, *theora*, *libtiff*, *toolame* и многие другие, что добавляет добавляет хлопот при сборке. Как говорится, всё своё несу с собой.

В итоге собрать *Cinelerra* из исходных текстов довольно сложно – выползают ошибки то в одной, то в другой библиотеке, включенной в дистрибутив программы. Однако, находятся смельчаки, собирающие пакеты. Эти пакеты разнятся по своей функциональности (подозреваю, потому, что сборщики пакетов отключают то, что не хочет компилироваться). В итоге один пакет понимает такие-то форматы файлов, а другой их не понимает.

Рис. 1. Однако, заставка при запуске красивая!



Кроме того, обычная версия и CV несколько отличаются друг от друга. Поэтому при описании приемов работы я буду делать упор на те функции, которые есть в обеих версиях и не буду «заморачиваться» с форматами. Допустим, *Cinelerra* по сути должна работать с видео, где есть альфа-каналы. Грубо говоря, это фильмы, где фон прозрачный и его не надо искусственно удалять, ежели в том возникнет нужда. Но в установленной у меня сборке *Cinelerra* видео с альфа-каналами не поддерживается. Я не буду говорить, что еще у меня не поддерживается, лучше отведем место для более полезного текста.

Эффекты и монтажные приемы мы будем разбирать на примерах. Я предполагаю, что вы обладаете начальными знаниями по работе с *Cinelerra* – то есть умеете загружать данные в проект, создавать новые дорожки, располагать на них фрагменты видео и для вас не составляет трудности и пользование некоторыми сторонними утилитами, например GIMP (читайте учебники предыдущих выпусков LXF!). А самое главное – у вас есть оцифрованные видеозаписи или статичные изображения для экспериментов. Почти всё, о чем я буду рассказывать, применимо как к видео, так и к обычным картинкам (TIFF, PNG и так далее) – с точки зрения *Cinelerra* безразлично, что вы монтируете.

Эффект замещения фона

Этот эффект очень любят на телевидении. Самый очевидный пример его использования – прогноз погоды. Ведущую (или ведущего) снимают на однородном и одноцветном фоне. Чаще всего это эдакий глубоко-синий (ближе к голубому) или салатовый (ближе к светло-зеленому, чем к желтому). Затем с помощью специального эффекта фон заменяется на изображение карты с температурами и всякими осадками. В любительском кино (вероятно, вы снимаете именно его) вам будет физически трудно снять актера на однородном и одноцветном фоне. Для этого нужны приспособления, хотя бы цветной «экран» и правильное освещение.

Однако эффекту есть много других применений. Допустим, вы создали в *Blender* трехмерную анимацию (см. LXF91) и хотите наложить ее на какое-то видео. Для этого вам надо вывести анимация на – да-да – одноцветном однородном фоне, и заняться в *Cinelerra* тем, что в монтаже называется по-разному. Иногда это “keying”. А иногда – “chroma key”. В русскоязычной среде можно встретить кальку «кеинг», хотя в английском языке “key” произносится как «ки», а не «кей», как некоторые думают и говорят.

Пара слов о цвете фона. Я уже говорил о двух основных – зеленом и синем. Такие цвета не имеют ничего общего с цветом человеческой кожи. Разумеется, актер или диктор должны избегать этих же цветов в одежде. И в обстановке тоже. При съемке цифровой камерой предпочтителен зеленый цвет. Иногда используют другие цвета – например, в «Матрице» это был малиновый, а в «Хищнике» – красный. И

если бы Киану Ривз одел малиновые штаны, то при «кеинге» малинового фона оказался бы без оных. И вообще его ноги стали бы невидимыми, хотя и в ботинках.

Для примера в этой статье я буду использовать видео, где на ярко-зеленом фоне движется динозавр. У меня в запасе есть и видео с некой девушкой на зеленом фоне, но я не уверен в законности его использования в качестве иллюстративного материала. Поэтому будем пользоваться динозавром. Модель была сделана Айвеном Рамаданом и помещена на www.3dkingdom.org для свободного скачивания. Я отрисовал ее в *3dMax* затем перевел в DV AVI (правильнее было бы вывести в несжатый AVI, но по иронии судьбы моя версия *Cinelerra* не понимает несжатый AVI).

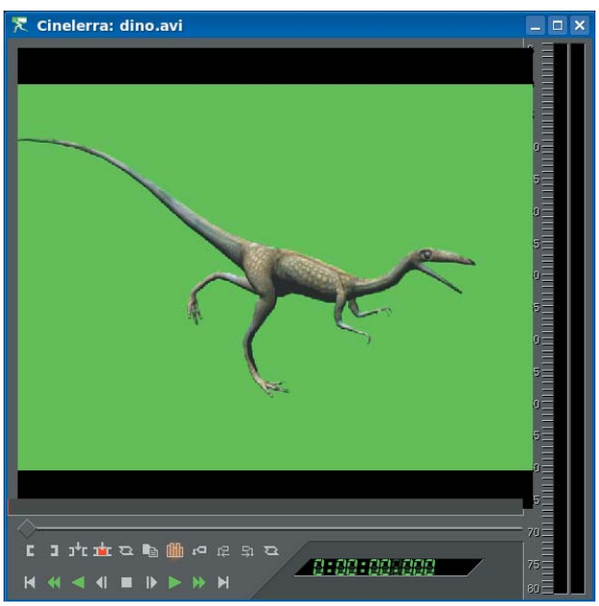


Рис. 2. Вот с таким динозавром нам придется иметь дело.

Для фоновой подложки – то есть того, на чем мы будем размещать динозавра – я взял собственноручно сделанную фотографию одной из киевских улиц. Она безлюдна – вероятно, прохожие разбежались при виде такого монстра.

Загрузите в *Cinelerra* подложку и видео (либо последовательность статичных кадров), снятое на зеленом или синем фоне (далее будем писать: фон и клип с динозавром) и расставьте их по дорожкам. Дорожка с динозавром должна быть над дорожкой с фоном.

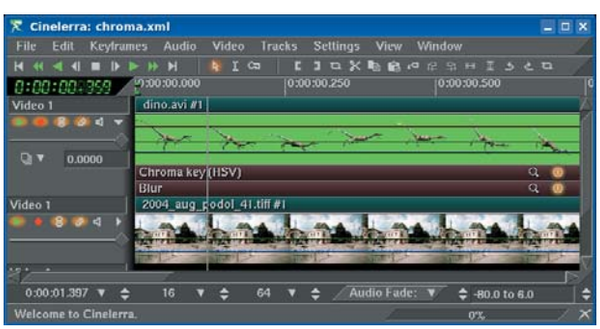


Рис. 3. Должно получиться примерно так. Обратите внимание, что на картинке для дорожки с динозавром уже помещены эффекты Chroma Key (HSV) и Blur.

Теперь отметим маркерами границы клипа с динозавром. Тащим на этот клип эффект... Постойте, в *Cinelerra* есть две эффекта, связанных с Chroma Key. Это просто Chroma Key и Chroma Key (HSV). Какой использовать? Разница между этими двумя заключена в способе, которым вы задаете программе, по каким критериям убирать фон. В “обычном” Chroma Key меньше параметров.

В нем вы можете:

1. Выбрать цвет фона, который надо удалить (нажмите кнопку Color и выберите нужный цвет). Как более точный вариант, в окне Compositor с помощью пипетки выберите цвет фона, а затем в окне эффекта Chroma Key нажмите кнопку Use Color Picker, чтобы «заполучить» в это окно выбранный в Compositor’е цвет. Эта секция настроек одинакова и в Chroma Key, и в Chroma Key (HSV) (**Рис. 4**).
2. Далее, у вас есть два ползунка – Slope и Threshold. Увеличение порога (threshold) убирает цвет в большем диапазоне, чем изначальный выбранный. То есть, чем больше значение порога, тем больше зеленого оттенка удаляется из изображения (удаленные пиксели становятся прозрачными). И напротив, уменьшение порога оставляет в кадре больше зеленых цветов. А ползунок Slope (откос, наклон) предназначен для сглаживания границ объекта, который расположен на фоне. В нашем случае этот объект – динозавр. Slope плохо работает с форматами видео, где применяется сжатие с потерями.

В связи с применением параметра Threshold уместно поговорить о выборе фонового цвета для объекта. Например, у вас есть синий автомобиль на голубом фоне. Очевидно, что регулируя ползунок Threshold, добиться четких границ автомобиля будет очень трудно – машина постоянно будет окружена фоновыми пикселями. Стало быть, фон для такой машины надо сделать зеленым, тогда его проще будет убрать. Итак, для фона выбираем тот цвет, который меньше всего присутствует в окраске объекта.

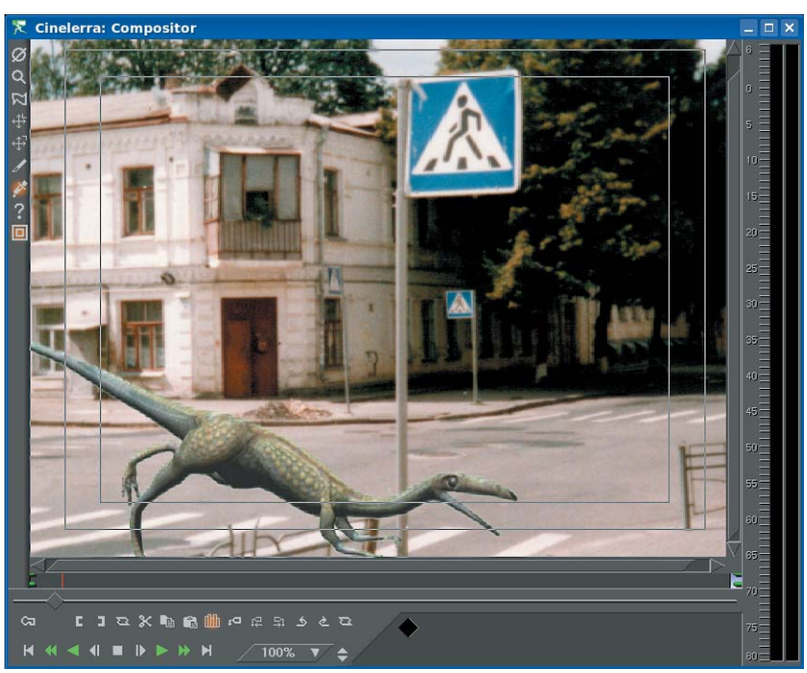
Теперь о Chroma Color (HSV). Работает этот эффект только если цветовая модель вашего видеопроекта является одной из следующих: RGBA8, RGBAFloat, YUVA8. Это те модели, в которых есть альфа-канал. Для изменения цветовой модели воспользуйтесь списком Color Mode в окне, вызываемом из Settings > Format.

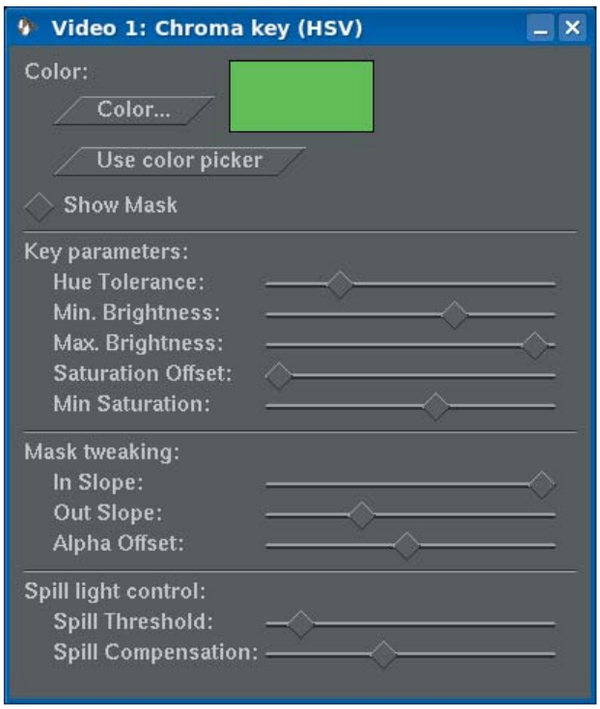
В Chroma Color (HSV) вы работаете с оттенком выбранного цвета (hue). Параметр Hue Tolerance позволяет управлять удалением цветов, сходных с выбранным. Поясню на отвлеченном примере. Допустим, вы сняли объект на фоне. Однако цвет этого фона получился не совсем однородным - на фоновый «экран» могла упасть тень, или наоборот, участок фона осветлился. Управляя значением Hue Tolerance, вы можете расширить для эффекта диапазон цветов, которые надо убрать, то есть охватить и тени, и светлые пятна.

Chroma Color (HSV) позволяет влиять на альфа-канал. Все ваши изменения параметров передаются маске, представляющей объект на альфа-канале. Чтобы посмотреть маску, поставьте галочку на Show

Рис. 4. Это и есть та самая пипетка в окне Compositor. Я довольно криво пометил ее красным кружком.

Рис. 5. В окне Compositor смотрим, что получается. Ура, зеленый фон динозавра чудесным образом исчез!





› Рис. 6. Окно параметров эффекта во всем великолепии.

» mask. Учтите, что маска передается в Compositor, то есть галочка на Show mask – это не просто способ видеть, как выглядит маска. Нет, маска станет частью кадра. Таким образом вы можете получать эффект движущегося силуэта. Некогда подобное было в заставке передачи «В мире животных», не знаю как теперь – я телевизор не смотрю (**Рис. 7**).

А если применить к такой маске эффект Invert Video, то получится другой любопытный эффект – прозрачный силуэт станет двигаться на однородном фоне, а подложка будет просматриваться в силуэте.

В Chroma Color (HSV) есть и другие настройки. Если границы объекта идут «лесенкой» из-за того, что его часть была расценена эффектом как фон, увеличьте значение Min Brightness – это может помочь. Если некоторые части объекта стали прозрачными, попробуйте уменьшить Max Brightness.

Скорая помощь

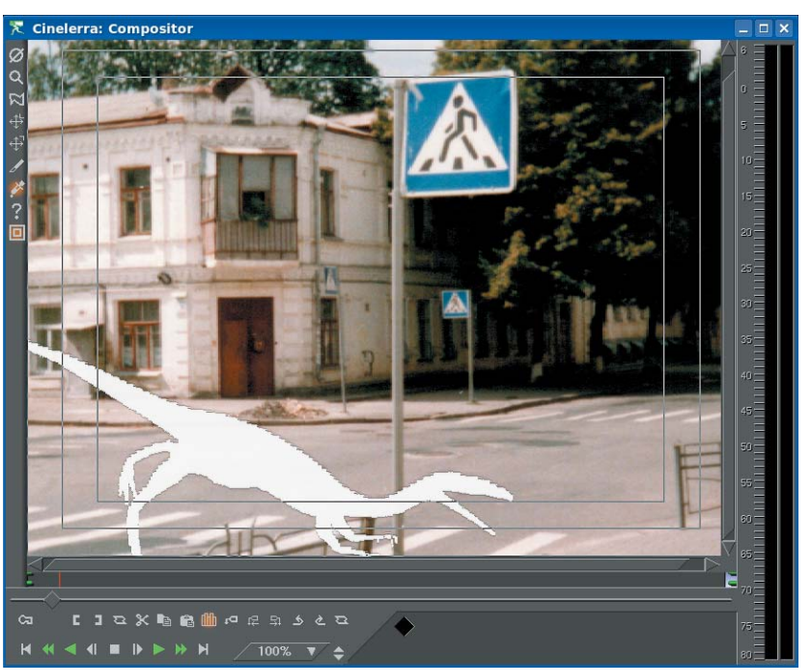
Если границы вашего объекта получаются «гребенкой» и никакие ухищрения не помогают, то попробуйте поставить эффект Blur (размытие) под Chroma Color (HSV), и в настройках Blur поставить галочку на Blur Alpha, отключив размытие для остальных каналов (Red, Green, Blue). А радиус размытия выставьте в 2 или 3.

Надо сказать, что в том же *Adobe Premiere Pro 2*, в аналогичном эффекте, настроек гораздо меньше. Однако, с другой стороны, там больше сходных эффектов – надо полагать, они несут в себе узкую специализацию, а Chroma Color (HSV) – это инструмент широкого профиля, универсальный.

Рассмотренному сегодня эффекту можно найти много применений – Chroma key используется не только в прогнозах погоды. Кроме динозавров и прочей живности есть еще космические корабли, летающие тарелки, взрывы, пламя (последние быстро и просто можно делать в *Adobe After Effects* и *Particle Illusion*), анимированные логотипы.

Chroma key – эффект, который вы найдете не в каждой монтажной программе, и его редко встретишь в ПО для новичков. Хотя использование данного эффекта простое, оно требует исходных видеоматериалов, которых у неподготовленного пользователя, скорее всего, нет под рукой. Ведь эффект работает, когда есть с чем работать. LXF

› Рис. 7. Кто облил динозавра краской?



» **Через месяц** Камеры, проекторы, маски – все, что необходимо для серьезного применения *Cinelerra*.

Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

Сегодня мы ответим на вопросы о:

1. fsck
2. Планештам и xorg.conf
3. Проблемах экрана
4. MP3
5. Сканерах
6. Мульти-загрузочных дисках
7. Kb3
8. Инсталляции Wine и OpenSUSE
9. RAM
10. SSH
11. BlueZ
- ★ RAID
- ★ Active Directory

1 Монтируя Mepis

Успешно заменив на своем компьютере старую систему Mandrake Linux на дистрибутив SimplyMepis 3.4 с диска LXF79, я нашел, что все работает безупречно, и перестал пользоваться F2 во время загрузки (ведь никаких неполадок не было). Однако несколько недель спустя я заглянул в загрузочные сообщения и увидел уведомление о том, что не смогла запуститься *fsck*, так как корневая директория не была смонтирована в режиме «только для чтения». В Сети подходящего объяснения я не нашел, и очень обрадовался, когда увидел учебник *Grub* в LXF90. Но он лишь сгустил туман. На моей машине, **/boot/Grub/menu.lst.example** содержит:

```
color cyan/blue white/blue
foreground ffffff
background 2f5178
gfxmenu /boot/grub/message
title MEPIS at hda2, kernel 2.6
kernel (hd0,2)/boot/vmlinuz-2.6.12-586tsc root=/dev/hda2 nomce psmouse.proto=imps
splash=verbose vga=791
initrd (hd0,2)/boot/initrd.img-2.6.12-586tsc
```

Однако **/boot/Grub/menu.lst** содержит:

```
color cyan/blue white/blue
foreground ffffff
background 0639a1
gfxmenu /boot/grub/message

title MEPIS at hda6, kernel 2.6.15-1-586tsc
kernel /boot/vmlinuz-2.6.15-1-586tsc root=/dev/hda6 nomce quiet vga=791
```

где формат кода загрузки ядра 2.6.15 не соответствует ни файлу-образцу, ни коду из учебника LXF90.

Хотя ОС, несмотря на это предупреждение, все же работает, мне, естественно, хотелось бы видеть *fsck* работающей, как полагается, и я был бы очень благодарен за любое предположение о причинах моей проблемы и о путях ее преодоления.

Питер Ненкерроу [Peter Nancarrow]

Файловые системы ext2/3 запускают *fsck* по наступлении заданного времени или количества загрузок; управляет этим *tune2fs*. Похоже, проблема возникла с первого дня, просто Вы еще не успели упереться в лимит, когда перестали читать загрузочные сообщения.

Между настройкой Вашего *Grub* и файлом-образцом есть два главных различия. Первое – Ваш загрузчик не пользуется *initrd* для отображения экрана-заставки, Ваша проблема тут ни при чем. Второе – для загрузки Mepis не указан корневой раздел. Если строка ядра не начинается с kernel (hd0,5)/boot/vmlinuz, Вас ожидают непредсказуемые последствия. Вариант – поместить root (hd0,5) в начало файла. Это не повлияет на проверку файловой системы при загрузке, но поможет избежать более тонких проблем.

Можно настроить *Grub* на монтирование корневого раздела в режиме «только для чтения», добавив го к строке параметров ядра. Файловая система перемонтируется согласно настройке из **/etc/fstab** на ранней стадии загрузки, но после того, как *fsck* завершит работу. Строка ядра будет выглядеть примерно так:

```
kernel (hd0,5)/boot/vmlinuz-2.6.15-1-586tsc root=/dev/hda6 ro nomce quiet vga=791
```

Можно запустить *fsck* вручную, загружаясь в режиме минимального вмешательства. При появлении меню *Grub* выберите пункт Mepis, нажмите E для его редактирования, выберите строку ядра и нажмите E снова. Удалите quiet и замените его на ro init=/bin/sh. Нажмите Enter для подтверждения и B для загрузки. Вы получите приглашение командной строки и, так как файловая система смонтирована в режиме «только для чтения», сможете запустить

```
fsck -f /dev/hda6 && shutdown -r -n now
```

Диск будет проверен, причем компьютер перезагрузится лишь в случае успешной проверки. **НБ**

2 Планшет блуждает

С моим планшетом Wacom Graphire3 творится что-то непонятное: при каждом включении машины приходится менять event в **xorg.conf**. Сейчас у меня Fedora 6 с Gnome, а мой **xorg.conf** унаследован от Fedora 5, с тем же Gnome и теми же аппаратными настройками. Раньше планшет работал безукоризненно. Поиск в Google результата не дал, и я понятия не имею, что происходит. Было бы здорово, если бы вы хоть что-нибудь мне подсказали.

Фред Капферрот [Fred Kupferroth]

Насколько я понял, Вам приходится всякий раз менять номер устройства event. Присланный Вами **xorg.conf** содержит

```
Option "Device" "/dev/input/event2"
```

и Вы вынуждены менять этот номер. А дело в том, что устройства ввода нумеруются в порядке их обнаружения, и этот порядок оказывается новым при каждой »

Наши эксперты

» Мы найдем эксперта для ответа на любой вопрос, от инсталляции и установки модемов до администрирования сети – лишь черкните письмо, а об остальном мы позаботимся.

Нейл Ботвик
Поруководив небольшим интернет-провайдером и поделав сопроводительные диски для LXF, Нейл считает себя «Linux-мастером на все руки».

Майк Сондерс
Майк стоял у истоков Linux Answers в LXF. Его специализация – программирование, оконные менеджеры, init-скрипты и SNES.

Пол Хадсон
Пол – местный супер-программист. Он управится со всеми вашими проблемами по части web и баз данных.

Ник Вейч
Когда не редактирует что-либо большим красным карандашом, Ник успевает разъяснить читателям все, что они хотят знать о графическом и 3D ПО для Linux. Готов ответить даже на самый простой вопрос!

Владимир Иванов
Работает с Linux уже более 10 лет, днем – создавая комплексные решения для крупных компаний, а вечером – модерируя сайт linux.org.ru и отвечая на вопросы о системном администрировании и информационной безопасности.

Валентин Синицын
В свободное от работы время редактор нашего журнала разрабатывает *KNetworkManager* и другие открытые приложения. Он с радостью поможет вам в вопросах использования Linux на рабочем столе.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите на м по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

загрузке – видимо, из-за какого-то устройства, подключаемого эпизодически (флэш-драйв или сканер). Выход – приказать *udev*, менеджеру устройств, присвоить Вашему планшету постоянное имя, не зависящее от порядка обнаружения.

Для этого существуют правила *udev*. Сначала выясните, как система определяет устройство, вот таким способом:

```
udevinfo -a -p /sys/class/input/event2 | less
```

Когда я подключаю свой планшет Aiptek, третий блок вывода содержит:

```
SUBSYSTEMS==”usb”
DRIVERS==”aiptek”
ATTRS{vendor}==”AIPTEK”
```

Конкретную запись привести невозможно (слишком много вариантов), но для своего Wacom-планшета Вы должны увидеть нечто подобное.

Wacom Technology Corporation

Благодаря udev ваш планшет обнаружится при каждой загрузке, невзирая на другие устройства.

Чтобы смастерить из этого правило *udev*, откройте терминал, станьте суперпользователем через *su* и отредактируйте файл **/etc/udev/rules.d/10/local.rules** (если такого нет, создайте его). И, пожалуйста, не поддавайтесь искушению добавить правило в уже существующий файл правил: при обновлении *udev* он может быть переписан. **10/local.rules** – самое место для ваших личных правил.

Теперь добавьте строку вроде этой, но со значениями, выданными Вам *udevinfo*:

```
SUBSYSTEMS==”usb”, DRIVERS==”aiptek”, ATTRS{vendor}==”AIPTEK”,    SYMLINK:=”input/tablet”
```

Как видите, здесь просто через запятую указаны атрибуты устройства, а затем идет установка SYMLINK.

Заметьте, атрибутам соответствует знак ==, означающий равенство, а в последнем пункте стоит :=, то есть значение присваивается. Ваше устройство будет создаваться как **/dev/input/eventN**, но иметь ссылкой **/dev/input/tablet**, запуск *ls -l /dev/input* это подтвердит. Теперь можете спокойно использовать **/dev/input/tablet** в **xorg.conf**, ваш планшет всегда будет работать.

Глубже разобраться в правилах помогут урок по *udev* в LXF66 (хотя кое-что с тех пор могло измениться, ведь *udev* непрерывно развивается) и полезный онлайн-учебник на **www.reactivated.net/udevrules.php**. **НБ**

3 Экран-уродец

В Неделю назад я купил ноутбук Dell Latitude (Pentium III) на распродаже излишков в местном университете. Прежде на нем стояла Windows, но продавался он без ПО, и мы решили попробовать Linux. Взяли журнал Getting Started with

Неказисто, зато безотказно: текстовая инсталляция Fedora Core.

Fedora Core 6 Linux [Спецвыпуск Linux Format], попробовали загрузить Fedora, и, после начальных загрузочных строк, получили прилагаемый экран – вместо того, что в журнале.

Джон Шнайдер [John Schneider]

О По присланному вами снимку экрана понятно, что инсталлятор Fedora Core неверно определил данные вашего дисплея, оттого экран и искажен. Есть несколько вариантов устранения проблемы. При появлении заставки, попробуйте набрать следующее:

```
linux skipddc
```

Инсталлятор не станет «вникать в детали» вашего экрана, а использует (надеемся, разумные) стандартные значения. Если это не поможет, попробуйте указать одно из следующих разрешений экрана:

```
linux resolution=1024x768
linux resolution=800x600
linux resolution=640x480
```

Если не поможет и это, можно запустить инсталлятор в текстовом режиме:

```
linux text
```

Вы получите неказистый, но полноценный установщик, управляемый клавишами курсора, Tab и Enter. Хочу подчеркнуть, что проблема дисплея касается лишь инсталляции, это не помешает вам установить и настроить графический рабочий стол. **НБ**

4 Извлечь звук

В Посоветуйте, какой цифровой формат выбрать для извлечения музыки с аудио-CD, чтобы можно было слушать ее без проблем с патентами? И как пакетно конвертировать сразу 30 альбомов – я перевел их в MP3 пару лет назад, будучи Microsoft-пользователем?

pootman, с форумов LXF

О Ogg Vorbis обеспечивает более плотное сжатие при лучшем качестве, чем MP3, и при этом совершенно свободен. Другой свободный формат – это FLAC (Free Lossless Audio Codec, свободный аудиокодек без потерь). Потерь здесь нет, так что экономии места ждать не приходится, зато сохраняется каждый бит оригинальной дорожки. Если места на жестком диске хватает, для хранения CD-дорожек лучше использовать FLAC – позднее Вы сможете конвертировать записи в Ogg Vorbis или MP3, чтобы перенести на малое устройство (например, флэш MP3-плейер).

Перекодирование готовых MP3 в Ogg Vorbis сопровождается некоторой потерей качества, поэтому лучше снова начать с CD. Для этого есть много графических инструментов, а проще всего использовать *Konqueror* из KDE. Поместите CD в привод и наберите media:/ в адресной строке. Выберите в списке CD, и Вы увидите его содержимое в виде файлов MP3, Ogg

Краткая справка по...

...запуску нескольких команд

Одной строкой можно запустить несколько команд: да или нет?

При работе в командной строке часто возникает необходимость последовательного запуска нескольких команд. Классический пример – компиляция ПО из исходных текстов, когда запускается **./configure** (возможно, с аргументами), а потом **make** и **make install**.

Каждый из этих последовательных шагов занимает от нескольких минут до нескольких часов, в зависимости от сложности кода и производительности машины. Дожидаться окончания одного, чтобы перейти к другому, неэффективно, поэтому можно сделать так:

```
./configure; make; make install
```

Точка с запятой вызывает последовательное выполнение команд, как будто их запустили по очереди.

Вы наверняка углядели потенциальную проблему: если **./configure** или **make** дадут сбой, последующие команды все равно выполнятся? Ответ – да: сообщения об ошибках будут скрыты, и вы ничего даже не заподозрите, пока не запустите программу.

Безопаснее связывать команды так:

```
./configure && make && make install
```

&& – это логический оператор. Строка фактически означает следующее: “if **./configure** is true and **make** is true and **make install** is true” [«если **./configure** истинно и **make** истинно и **make install** истинно»]. К счастью, оболочка определяет истинность команды ее запуском и наблюдением за ошибками. В случае отказа команды необходимость в запуске следующей отпадает, так как тест уже провален, поэтому объединение команд через && запускает их последовательно, но прекращает работу при первом же сообщении об ошибке (а значит, больше незачем караулить оболочку) [фактически, проверяются не сообщения об ошибках как таковые, а код возврата команды, – прим.ред.].

Компаньон команды && – это ||, что значит ‘or’ (или). Так, в выражении

```
command1 || command2
```

command2 работает только в случае провала command1. В интерактивной оболочке это особо ни к чему, но часто используется в скриптах:

```
somecommand1 || echo “Что-то рухнуло!”
```

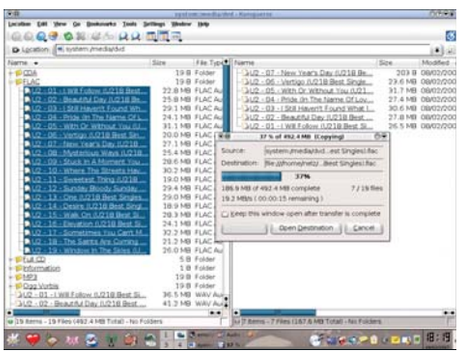
» Vorbis, FLAC и Wave. Понятно, все эти файлы вообразимые, но при копировании их на жесткий диск они будут перекодированы «на лету». Параметры сжатия устанавливаются в разделе Sounds & Multimedia (Звуки и Мультимедиа) Центра управления KDE.

Если Вы не пользуетесь KDE, рекомендую *Grip* с **www.nostatic.org/grip**. Это GTK-аудиоплейер и ripper. Для консоли есть abcde (с **www.hispalinux.es/~data/abcde.php**), сценарий оболочки: он извлечет и перекодирует файлы и снабдит их тэгами. Он идеален для конвертации партии дисков – только успевайте подбрасывать новые взамен обработанных. Все указанные методы используют онлайн-базу CDDB для добавления справочной информации к создаваемым файлам.

При наличии только MP3 поможет скрипт *mp32ogg*, взять который можно на **http://faceprint.com/code**. Пользоваться им несложно:

```
mp32ogg musicdir
```

и все файлы из **musicdir** будут перекодированы из MP3 в Ogg Vorbis. Различные параметры управляют уровнями качества, именованием файлов и удалением оригиналов – запустите *mp32ogg --help* для их просмотра. Этот скрипт не только конвертирует музыку, но и переносит тэги из MP3-файлов. НБ



› **В *Konqueror* кодировать дорожки аудиодиска в Ogg Vorbis, MP3 или FLAC не сложнее, чем копировать файлы.**

5 Сканирование – народу

В Недавно я перешел с SUSE 10.0 на OpenSUSE 10.2 через реинсталляцию. Мой сканер CanoScan N640P хорошо работал в SUSE 10.0 с *XSane/SANE*, работает и теперь, но лишь с привилегиями администратора. Поддержка *SANE* для сканера надежна. Версии *XSane/SANE* – те, что были в дистрибутиве, т.е. 0.991-32 i586 и 1.0.18-34 i586. Вскоре после инсталляции я сделал онлайн-обновление. Версия *SANE* на SUSE 10.0 была 1.0.15-20.2 i586.

Yast2 корректно настроил сканер – вручную было добавлено canon_pp: parport0 – и протестировал со *scanimage -d canon_pp:parport0 -T*. Я успешно запускаю эту команду из командной строки, но только как суперпользователь.

Вызывая *XSane* как обычный пользователь, я получаю сообщение: 'no devices available' (нет доступных устройств), с указанием шести возможных причин. Из них третья, 'the permissions of the device file do not allow you to use it – try root' [недостаточно прав – попробуйте как суперпользователь], кажется наиболее вероятной. Однако я не могу найти файла устройства для сканера. Хотя, насколько я могу судить, файлы конфигурации верны.

Пробовал все четыре настройки параллельного порта в BIOS (normal, ECP (DMA3), ECP/EPP и EPP), но безрезультатно.

Дейв Кулсток [Dave Coulstock]

О Вы правы, все дело в правах доступа. То же самое было у меня с USB-сканером Canon. Устройство «сканер» – это **/dev/parport0** (хотя некоторые дистрибутивы называют его **/dev/lp0**).

Запуск

```
ls -l /dev/{par,lp}*
```

покажет все эти устройства и права доступа к ним. Обычно выводится что-то вроде:

'crw-rw---- 1 root lp 99, 0 Jan 27 11:37 /dev/parport0'

Это показывает, что устройство доступно лишь суперпользователю и членам группы **lp**. В таком случае проще всего добавить себя в группу **lp**, командой

```
gpasswd -a yourusername lp
```

Некоторые дистрибутивы создают группу **scanner** вместо **lp** – тогда команду, естественно, придется изменить. Чтобы изменения заработали, выйдите из системы и войдите снова. Затем попробуйте

```
scanimage --list-devices
```

как супер- и как обычный пользователь. В обоих случаях сканер должен быть виден.

Если устройство принадлежит root:root или права доступа не rw-rw----, Вам придется изменить ситуацию, создав правило *udev*. Добавьте следующую строку в **/etc/udev/rules.d/10/local.rules**:

```
KERNEL==”parport0”, GROUP:=”scanner”,
MODE:=”660”, SYMLINK:=”scanner”
```

Она предпишет устройству **/dev/parport0** иметь права доступа rw-rw---- и принадлежать к группе scanner. Создается также символическая ссылка **/dev/scanner**, которую «уважают» некоторые программы. Подробнее о создании правил *udev* см. ответ на вопрос «Планшет блуждает», стр. 94.

Если Ваше устройство не **parport0** или **lp0**, найти его можно так:

```
dmesg | grep -i -C 3 -e parport -e canon -e sane
```

НБ

Часто задаваемые вопросы...

Live CD

Любимцы мира дистрибутивов, они так малы, что их можно всюду носить с собой – но не ждите от них слишком многого…

» Что такое Live CD?
Это дистрибутив на одном CD (DVD), работающий прямо с диска.

» Разве не все диски Linux-дистрибутивов загрузочные? Чем отличается Live CD?
Обычный диск загружается в инсталлятор, и нужно установить дистрибутив на жесткий диск и работать оттуда. Дистрибутив Live CD работает прямо с компакта, в большинстве случаев загружаясь прямо в рабочий стол. Инсталлировать нечего: загрузился – и работай.

» А можно сохранить результаты своего труда под Live CD?
Не на CD, конечно – он доступен лишь для чтения. Однако большинство Live CD дают возможность сохранить работу и настройки, чаще всего помещением домашней директории на USB-брелок, так что вся работа и настройки хранятся там. Альтернатива – дискеты или разделы винчестера. Некоторые Live CD проделывают ловкий фокус: вам кажется, что вы пишете данные обратно на CD, а на самом деле они записываются в специальный файл на жестком диске или USB-носителе. Преимущество такого способа в том, что можно менять любой файл, не только домашнюю директорию. Можно даже инсталлировать новое ПО.

» В чем преимущество Live CD?
Их можно запускать везде, на любом компьютере, загружающемся с CD. Хотите купить новый ноутбук? Загрузитесь с Knoppix CD и убедитесь, что его оборудование опознано и настроено: мгновенный тест на Linux-совместимость. Такие диски полезны также для восстановления после катастрофы или для операций обслуживания. Если случайно поврежден критический файл и компьютер не загружается, загрузитесь с Live CD и исправьте ошибку. Хотите скопировать содержимое винчестера на больший диск – подключайте оба, загружайтесь с Knoppix для подготовки места – и спокойно копируйте свои данные.

» А каковы недостатки?
Самый очевидный – ограниченные возможности для записи и потребность во втором устройстве для хранения созданных файлов. Другой недостаток – по сравнению с обычной инсталляцией, Live CD медлительны. Эквивалент жесткого диска хранится в большом сжатом файле на CD, обычно содержащем около 2 ГБ упакованных файлов в объеме меньше 700 ГБ. Скорость доступа и скорость чтения у CD ниже, чем у жестких дисков, да еще нужно распаковать данные. Время загрузки программ увеличивается, но, единожды загрузившись, они работают, как всегда, быстро.

» Как насчет DVD?
Большинство Live-дистрибутивов распространяется на CD, но все больше проектов предлагают DVD-альтернативы, дающие ощутимый прирост в количестве доступного ПО.

» Где можно найти Live CD?
На наших сопроводительных дисках, конечно. Существуют внушительные списки Live CD на **www.livecdlist.com** и **www.linuxlinks.com/Distributions/LiveCD**.

6 Грузятся диски...

Я почитываю ваш журнал уже года полтора. Браво! А трехзагрузочный DVD из LXF79 вообще крут, как вареное яйцо. Теперь вопрос: как вы это делаете? Хочу смастерить нечто подобное для друзей и знакомых, чтобы они «почувствовали Linux-разницу». Я приобрел пишущий DVD и диски, так что с «железом» все в порядке. Осталось узнать, как записать несколько дистрибутивов и сделать каждый из них загружаемым. Куча благодарностей заранее. Буду читать вас и далее, если удержите марку!

Марк [Mark]

Сделать мультизагрузочный DVD непросто: нужно изучить загрузчики *Grub* и *Isolinux*, а также структуру тех дистрибутивов, которые Вы хотите скомбинировать. Сначала создайте новую директорию и скопируйте в нее содержимое первого CD- дистрибутива. Если Вы не прожгли CD, то можете смонтировать ISO-образ как loopback-устройство (от имени суперпользователя):

```
mkdir /loop/
mount -o loop discimage.iso /loop/
```

Скопировав содержимое диска в новую директорию, сделайте то же для второго дистрибутива. Просмотрите файлы: не будет ли конфликтов с первым дистрибутивом? Если есть, понадобится вручную отрихтовать (возможно, даже пересобрать) дистрибутив, придется попотеть. Если конфликтов нет, или они ограничиваются директориями **grub** и **isolinux**, то все в порядке.

Скопируйте файлы второго диска в новую директорию. Теперь в ней содержится два дистрибутива. Затем нужно настроить загрузчик на мультизагрузку. Выберите загрузчик одного из дистрибутивов (в директории **boot**, **grub** или **isolinux** – зависит от дистрибутива) и отредактируйте его файлы конфигурации (**menu.lst**, **isolinux.conf** – см. документацию *Grub* и *Isolinux*, там указаны типичные имена файлов).

Хотите создать мультизагрузочный диск вроде нашего – во всем слушайте Майка: это дело может оказаться не из простых!

Добавьте в файл конфигурации загрузочные записи для второго дистрибутива: Вы найдете их в директории его загрузчика. Теперь файл первого загрузчика будет содержать загрузочные записи второго дистрибутива. Еще не запутались? Хоть это и нелегко, но следите за тем, чтобы не потерять записи из обоих конфигурационных файлов; затем объедините их.

Теперь прожгите директорию на диск, используя элемент директории загрузчика первого дистрибутива в качестве загрузочного блока – он будет называться **isolinux.bin** (для *Isolinux*) или **stage2_eltorito** (для *Grub*). Если Вы объединили файлы загрузчиков правильно, а директории дистрибутивов не конфликтуют, то можно будет загружаться с нового диска и выбирать нужный дистрибутив в загрузочном меню. **MC**

7 Жгучий вопрос

По работе я прожигаю диски с помощью *K3b*. При прожиге отметки времени для всех записанных файлов становятся одинаковыми. Я хотел бы сохранять оригинальные отметки времени, но не знаю, как это сделать. Вы не могли бы мне помочь?

fleafly, из форумов LXF

K3b вообще-то делает это по умолчанию, но данное поведение можно изменить. Открыв окно проекта *K3b*, после щелчка на кнопке Burn (Прожечь) перейдите на вкладку Filesystem (Файловая система) и убедитесь, что установлена галочка Preserve File Permissions (Backup) (Сохранять права доступа). Нужно отметить галочками также Generate Rock Ridge Extensions (Создавать расширения Rock Ridge) и Generate Joliet Extensions (Создавать расширения Joliet) для полной совместимости. Расставив все галочки, нажмите кнопку Save User Defaults (Использовать по умолчанию), и программа сохранит сделанные изменения.

Если запись CD производится постоянно, удобно будет написать короткий скрипт оболочки, что-нибудь вроде:

```
#!/bin/bash
SOURCE_DIR="~/work/data"
DVD_WRITER="/dev/dvd"
```

Вопрос-победитель (русская версия)

Павел Воинов получает подарочный сертификат на 1000 рублей от интернет-магазина LinuxCenter.Ru! Просим победителя выйти на связь с редакцией: info@linuxformat.ru

Свой-чужой

Фирма, в которой я работаю, недавно начала переход с Windows на Linux – подействовали множество обсуждений о том, что Linux готов к применению в качестве ОС для рабочих станций обычных пользователей. Но, к сожалению, первая попытка не увенчалась 100% успехом. Миграцию, естественно, было решено выполнять не в виде «шоковой терапии», а постепенно. В итоге на нескольких рабочих станциях был установлен OpenSUSE 10.2. Одними из причин выбора этого дистрибутива явились интеграция с существующей инфраструктурой ActiveDirectory «из коробки», поддержка макросов OpenOffice.org VBA (опираясь на информацию из вашего журнала).

Вот с первым-то и случились проблемы. С тем, чтобы настроить авторизацию пользователей в OpenSUSE при помощи Samba (установив членство в существующем домене), действительно, никаких трудностей не возникло. Пользователи могли входить под своими «доменными» учетными записями. Но при этом им не предоставлялся доступ к локальным устройствам – плееры (в частности, *Amarok*) завершались ошибкой «xine не может обнаружить аудио-драйверы», K3b отказывалась обнаруживать приводы, даже монтирование дисков DVD/CD и Flash приводило к завершению с ошибкой, связанной с нехваткой прав.

Все эти проблемы, как я выяснил, связаны с тем, что такие «доменные» пользователи не включены ни в одну локальную Linux-группу (audio, burning и др.). Если авторизоваться под любым локальным пользователем, то таких проблем не возникает. Единственное решение, которое я сейчас нашел – запускать эти программы с помощью sudo под специально созданными локальными пользователями, что, согласитесь, не совсем правильно.

Есть ли какой-нибудь способ, автоматически включать «доменных» пользователей при авторизации в нужные локальные группы? Как это можно сделать?

Павел Воинов

Вероятно, у Вас не настроен механизм приведения в соответствие групп пользователей Linux и Windows или пользователи не внесены в локальную базу данных групп. В зависимости от того, как производится преобразование символьных имен пользователей в численные идентификаторы, решение может отличаться.

Текущие настройки можно узнать в файле **/etc/nsswitch.conf**. Для проверки того, как работают функции преобразования для групп, введите:

```
# grep ^group /etc/nsswitch.conf
    group:          files
```

Если информация о группах хранится в файле, отредактируйте **/etc/group** и внесите соответствующих пользователей в группу **audio** (формат файла **/etc/group** подробно описан в man), например:

```
audio:x:29: bgates
```

Возможно, преобразование обеспечивается средствами Samba:

```
# grep ^group /etc/nsswitch.conf
group:          winbind
```

В этом случае, между доменными и локальными группами пользователей должно быть установлено соответствие. Это достигается с помощью команды net groupmap из пакета Samba, например:

```
# net groupmap add
ntgroup="Audio Users"
unixgroup=audio type=d
```

После этого, все пользователи, включенные в доменную группу «Audio Users», будут включены в локальную группу audio.

После того, как настройки произведены, проверить наличие пользователя в локальной группе можно следующей командой:

```
# getent group audio
    audio:x:29: bgates, sballmer
```

Приятного прослушивания!

ВИ

Вопрос-победитель (английская версия)

Загрузка RAID

У меня несколько месяцев отлично работает Debian Etch, но хотелось бы перенести его с **hdb** на программный RAID 1, состоящий из **hde** и **hdg**. Моя текущая конфигурация:

- **/boot** на **hdb1**, ФС ext2.
- **/** на **hdb2**, ФС ReiserFS.
- **/home** на **hdb3**, тоже ReiserFS.

Я могу перенести **/boot** на **hdb0**, **/** на **hdb1** и **/home** на **hdb2**, но как потом настроить *Grub*? Я знаю, что *Grub* ведет отсчет дисков и разделов от 0, но как быть с RAID? Как переустановить *Grub* для моего нового массива?

Harvey

О Вы удивитесь, как просто обращаться с RAID 1. Отдельные диски доступны как автономные, так как данные RAID не пишутся на корневой блок диска. Если **/dev/md0** собран из **/dev/hde1** и **/dev/hdg1**, то любой из них доступен как загрузочный диск. Вероятно, они будут названы (**hd1,0**) и (**hd2,0**), так как **/dev/hdb** будет (**hd0**), но это несложно проверить *Grub*'овской командой *find*.

```
touch /boot myraidbootdisks
grub
#переключение на приглашение grub
find /myraidbootdisks
```

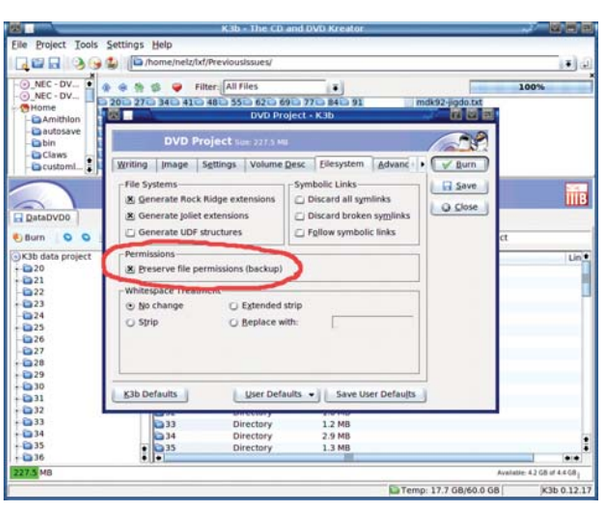
Вы увидите нечто вроде этого:

```
find /myraidbootdisks
(hd1,0)
(hd2,0)
```

Загружаться можно с любого из них, но я рекомендую сделать настройку *Grub* для каждого и вставить в меню два пункта, по одному для каждого диска. Тогда, если одна из загрузочных ФС рухнет, можно будет загрузиться с другого диска. Как только ядро загрузится и создастся массив RAID, он сам себя исправит.

В команде *find* можно использовать любой файл, но создание файла вместо поиска ядра устраняет возможность конфликтов со старой настройкой. Если Вы хотите удалить **hdb** после настройки *Grub*, не забудьте о том, что нумерация изменилась и файлы меню *Grub* нуждаются в правке. Если Вы установите *Grub* на оба диска, как рекомендовано выше, то первый пункт меню все же будет работать, хотя и будет обращаться ко второму диску. Это может слегка запутать, но все же позволит загрузиться и поправить файлы меню. ПХ

Grub обращается с RAID 1 как с обычными дисками, поэтому при настройке загрузчика нет нужды в особых действиях.



***Kb3* сохраняет отметки времени для файлов, если его хорошо попросить.**

```
ISO_FILE=”~/tmp/image.iso”
mkisofs -rdJ -o $ISO_FILE $SOURCE_DIR && cdrecord
dev=$DVD_WRITER $ISO_FILE
```

Вам понадобится новейшая версия *CDRecord* – или *wodim*, включающая версию *CDRecord*, которая умеет писать DVD – чтобы использовать ее с пишущим DVD-приводом, иначе измените последнюю строку так:

```
growisofs -dvd-compat -Z $DVD_WRITER -rdJ
$SOURCE_DIR
```

но помните, что *growisofs* работает только с DVD, а не с CD. MC

8 Горести инсталляции

В Я решил сделать следующий шаг в Linux: заставить работать с помощью *Wine* мой инженерный пакет расчета конечных элементов. Попробовал установить Ubuntu 6.10 с диска LXF88, но экран постоянно мигал во время загрузки, даже когда я использовал Start With Low Resolution (Запуск с низким разрешением) и с помощью F4 менял разрешение на поддерживаемое монитором и видеокартой. По-моему, частота обновления экрана моего монитора 50 Гц, но Ubuntu 6.10 и Fedora Core 6 устанавливают ее на 60 Гц, независимо от разрешения.

Тогда я попробовал Ubuntu 6.06 – вот он инсталлировался безупречно. Хотел установить *Wine* из исходных текстов – **./configure** предложила установить '*flex*', а тот предложил установить '*m4*', затем потребовался еще и '*Bison*'. И после всего этого *Wine* вернул сообщение об ошибке во время *make*.

Не справившись с Ubuntu, я попробовал установить Open SUSE 10.2 с DVD от LXF89, но посреди инсталляции получил такое сообщение:

```
“error occurred while creating the catalog
Cd///?devices=/dev/hdc source rejected by the user
Retry (yes) (no)”
```

Нажатие Yes принесло сообщение 'error occurred dvd/// source rejected by the user'. У меня на жестком диске была двойная инсталляция с Windows XP – может быть, в этом причина?

Анвар Ализаде [Anvar Alizadeh]

О Вы и впрямь прошли крещение огнем! Попробую разобраться в причинах ваших неудач. Большинство мониторов имеют минимальную частоту обновления экрана 60 Гц, но после инсталляции вы можете установить то, что подходит Вашему монитору, в **/etc/X11/xorg.conf**. Найдите ту часть, которая начинается Monitor: там вы увидите настройки для HorizSync и VertRefresh. Замените их данными из спецификации монитора.

Большинство дистрибутивов предоставляет широкий выбор ПО, предполагая, что обычные пользователи редко компилируют что-то из исходников, так что инструменты для сборки по умолчанию не устанавливаются. Для Ubuntu возможно два варианта решения проблемы *Wine*. Можно инсталлировать пакет *build-essentials*, и тогда вы получите все необходимое, включая *flex* и *m4*.

Но проще будет добавить хранилище WineHQ в свой список пакетов, а затем установить новейшую версию *Wine* через менеджер пакетов. Для добавления хранилища запустите следующие команды:

```
wget -q http://wine.budgetdedicated.com/apt/387EE263.
gpg -O- | sudo apt-key add -
sudo wget http://wine.budgetdedicated.com/apt/
sources.list.d/edgy.list winehq.list
-O/etc/apt/sources.list.d/winehq.list
```

Первая добавит ключ от хранилища в Ваш список доверенных ключей, а вторая добавит само хранилище. Этот вариант для Ubuntu 6.10; для 6.06 замените edgy на dapper во второй команде.

Инсталляторы дистрибутивов иногда не могут найти привод, с которого происходит установка. Обычно это происходит тогда, когда в системе два оптических привода: Вы загружаетесь с одного диска, а инсталлятор обнаруживает другой и пробует копировать данные с него. Тогда проще будет загрузиться с другого дисковода. Если это невозможно (например, первый привод – CD, а вы пользуетесь DVD), то временно отключите первый дисковод.

Физически отсоединять кабель не обязательно: большинство BIOS умеют отключать отдельные устройства. Эта проблема касается лишь инсталляции – по окончании ее можно подключить все, как было. Подобное иногда случается при попытке инстал- »

» ляции с DVD-приводов, подключенных по USB.

Возможно также, что Ваш диск поврежден. Проще всего проверить это на другом компьютере. Инсталлировать ничего не надо, просто загрузитесь и посмотрите, правильно ли работает инсталлятор. **НБ**

9 Что-то с памятью моей

Я в курсе, что объем памяти – узкое место моего сервера, уже потому, что сервер постоянно использует подкачку. Но точно определить количество памяти для эффективной работы оказалось не так-то просто. Я мог бы просто купить столько памяти, сколько мне по средствам, но мне кажется, что существуют какие-то методы определения оптимального количества.

Phandro

Ваш вопрос, на первый взгляд простой, требует для ответа глубокого знания Linux. Для начала, подкачка пусть Вас не беспокоит. Нормальная Linux-система практически постоянно использует swap-пространство [спорное утверждение: на наших ПК swap обычно простаивает, – прим.рус. ред.]. Хотя злоупотребление подкачкой, конечно, нежелательно.

Например, вы оснастили свою машину новейшим многоядерным процессором, а скорость не увеличилась: при интенсивном обмене данными, если оперативная память мала и работает медленно, сервер по большей части будет гонять данные туда-сюда, почти не занимаясь их обработкой.

На Linux немало инструментов, которые помогут разобраться, что, собственно, происходит на вашей машине: весьма полезны *top* и *uptime*. Наряду с прочими сведениями, *uptime* отображает три колонки цифр: среднюю нагрузку машины за последние минуту, пять и пятнадцать минут. Что значит «нагрузка»? А это такое число, отражающее количество работы, выполненное компьютером. Крупные цифры означают большую работу, мелкие – работу поменьше. По-настоящему это «скользящее среднее с экспоненциальным затуханием от длины очереди запросов на CPU»… уж лучше считать его просто волшебным числом.

От этого числа как такового пользы мало: машина может быть очень занятой, но прекрасно справляться с нагрузкой. А вот безудержный его рост свидетельствует о проблемах. Запуск *top* покажет все работающие процессы и распределение между ними ресурсов процессора и памяти. Но, как я уже сказал, плотное использование CPU не всегда плохо, а редкое – не обязательно хорошо. Последнее может указывать на то, что данные слишком долго входят в процесс и выходят из него, поэтому низкая производительность при высокой загрузке – это звоночек о том, что скорость I/O (ввода/вывода) недостаточна (купите WD Raptors и хороший контроллер), или недостаточно физической оперативной памяти.

Комбинация *top*, *uptime* и *free* (показывает использование памяти) поможет определить, что подходит именно вам.

Если вам некогда дожидаться высокой активности сервера для проверки, создайте эту активность сами. Например, в состав *Apache* входит тестер, *ApacheBench* (команда называется *ab*), способный имитировать бурную деятельность.

Может оказаться полезным просмотр процессов, запущенных в компьютере. Хотя под Linux службы запускаются лишь в случае необходимости, все же некоторый выигрыш можно получить, убив все «левые» процессы. Демоны печати, звуковые серверы, *X*… даже HAL тут лишний, однако часто работает по умолчанию.

Чтобы действительно выжать из своей памяти все, попытайтесь перекомпилировать программы, оставив только действительно необходимые вам функции. Можно получить ощутимое снижение потребления памяти *MySQL*, *Apache*, PHP и иже с ними.

Множество полезных сведений можно найти в Сети; рекомендую статью об анализе памяти Лубоса Лунака [Lubos Lunak] на **http://ktown.kde.org/~seli/memory**. **НВ**

10 Ужесточить SSH

Для моих задач нужно запускать удаленную оболочку (SSH, я полагаю) на моем сервере. Но я не люблю перегружать сервер лишними службами, и не уверен, что оставлять SSH-сервер работающим безопасно. Я не очень-то в этом разбираюсь, и хотел бы знать, как усилить безопасность.

Грег [Greg]

Хотя SSH безопасна по умолчанию, способы повысить устойчивость к взлому есть всегда. В основном они касаются ограничения методов входа, учетных записей и мест, с которых можно входить.

По умолчанию SSH допускает простой вход через пароль. При этом, соединяясь с сервером SSH как обычный пользователь, нужно этот пароль ввести. Конечно, пароль можно подобрать, так что есть и другие способы: например, авторизация в SSH через пару доверенных ключей. Потребуется генерация ключа на стороне клиента и копирование публичной части в хранилище authorized_keys (авторизированных ключей) SSH-сервера. Это удобный метод быстрого подключения без необходимости запоминать пароль, к тому же можно отключить запрос пароля на SSH-сервере. Сначала сделайте ключ и скопируйте его на сервер:

```
ssh-keygen -t dsa scp ~/.ssh/id_dsa.pub  servername:.ssh/authorized_keys2
```

Подразумевается, что вы входите с одним и тем же именем пользователя на обоих компьютерах.

Необходимо отредактировать файл **/etc/ssh/sshd_config**, изменив строку

```
PasswordAuthentication yes
```

на

```
PasswordAuthentication no
```

Убедитесь в том, что сможете войти со своим ключом, прежде чем пробовать это, особенно на удаленном сервере!

Пока файл открыт, можно попробовать еще пару трюков. Найдите следующие две строки (в файле они не соседствуют):

```
PermitRootLogin yes
...
Protocol 2,1
```

и замените их на:

```
PermitRootLogin no
Protocol 2
```

Это предотвратит прямой вход чужака от имени суперпользователя – для полного доступа нужно будет войти как обычный пользователь, а после стать суперпользователем через *su*. Причина для этого проста: любой потенциальный взломщик, вместо подбора одного пароля (или ключа, как в Вашем сверхпрочном варианте), будет вынужден подбирать два, да еще и знать имя пользователя – мороки куда больше.

Вторая строка заставляет пару сервер–клиент пользоваться более безопасным протоколом для SSH-соединения. Вряд ли есть клиент, его не поддерживающий, поэтому установите этот режим прямо сейчас!

Вдобавок к укреплению пользовательского входа, можно разрешить его только определенным пользователям, так как некоторые названия учетных записей на Linux-машине угадать нетрудно.

```
AllowUsers  eric jeff mike degville
```

Простой список, разделенный пробелами, разрешит доступ лишь указанным пользователям.

Если хотите быть реально крутым, «привяжите» учетные записи к определенным компьютерам, добавив доменное имя инициирующего сервера (будьте осторожны, доступ из некоторых источников не всегда может исходить из постоянного IP-адреса):

```
AllowUsers   mike@linuxformat.co.uk  eric@*.ac.uk
```

Уж это отвадит злоумышленников от SSH. **НВ**

11 Блюз BlueZ

Долгое время я пользовался *BlueZ 2.25* в Slackware и все работало, но на днях поставил Arch Linux с *BlueZ 3.x* и оказалось, что в этой новой версии перестал работать параметр passkey. На форумах пишут, что загвоздка в D-Bus, но документации-то нет! Может у меня версия сырая? Или надо пользоваться *BlueZ 2.25*?

Михаил Шагов

Это достаточно известная проблема. Все дело в том, что передачей PIN-кода в *BlueZ 2.x* обычно занималась некоторая сторонняя программа, известная как *pin_helper*, полный путь к которой и передавался в соответствующем параметре. Она просто запрашивала PIN у пользователя и просто выводила его в STDOUT. *BlueZ 3.x* сделал шаг вперед – и теперь не только работа с PIN, но и многое другое производится через D-Bus. С одной стороны, это хорошо (D-Bus – гораздо более гибкий механизм, специально созданный для этих целей), а с другой – пользовательские интерфейсы, которые не перешли на D-Bus (в первую очередь, *kdebluetooth*) не смогут полноценно работать с *BlueZ 3.x*. Вариантов два: либо откатиться обратно на 2.x, либо наложить на *BlueZ* патч, возвращающий функциональность *pin_helper*. Детали можно найти по адресу: **http://kmobiletools.org/node/228**. Кроме того, в состав *BlueZ 3.x* входит утилита *passkey* – можно воспользоваться ею напрямую:

```
passkey --default 1234
```

где 1234 – требуемый PIN-код. Утилиту необходимо держать запущенной в течение всего сеанса связи. **ВС** LXF

Лучшие новинки открытого ПО на планете

LXF HotPicks

В ЭТОТ РАЗ ТОЛЬКО ДЛЯ ВАС: Kdesvn » GREYCstoration » XML Copy Editor » Raw Studio » TuxWordSmith » UFO: AI » ClamTk » Nagios » KeyFrog » AcetoneISO

Ник Вейч
Чудо-математик, фанат Flickr, иногда редактор, Ник из высокой башни LXF орлиным взором оглядывает поле свободных программ в поисках симпатичных приложений.

Клиент Subversion

Kdesvn

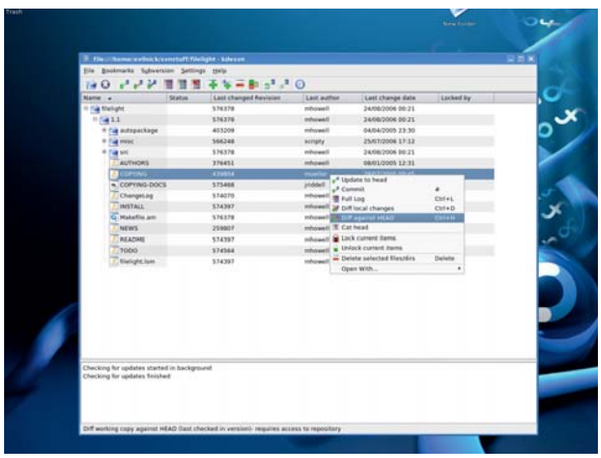
Версия 0.11.1 **Сайт** www.alwins-world.de/wiki/programs

Это поразительное 3D-приложение для Gnome делает... вы правы: на самом деле *Kdesvn* – клиент Subversion для KDE. Угадали. Ну и что, все равно ведь круто? Ладно, вы опять нас поймали: вряд ли кто-то сбежит с работы домой пораньше, чтобы проверить файлы журналов репозитория. Но все равно, функциональные инструменты вроде этого облегчают нашу жизнь и сберегают время на развлечения.

Реальные разработчики, естественно, пользуются командной строкой. Понятное дело. Для проектов, над которыми вы все время трудитесь, это, вероятно, самый эффективный подход (а еще лучше воспользоваться редактором или IDE, закачивающими для вас обновления). Но бывают моменты, когда нужно просто просмотреть репозиторий или глянуть на не слишком знакомый проект. И как приятно иметь удобное в использовании окружение, где можно это сделать. Честно.

Благодаря *Kdesvn*, подключиться к SVN-репозиторию не сложнее, чем нажать Ctrl+O и ввести URL в диалоге. Главная панель содержит дерево выбранного проекта; правый щелчок на любом элементе панели выводит список доступных опций, содержащих загрузку файлов, удаление целых каталогов (конечно, если у вас есть на это права) и проверку журнала работ. Последнее выводит другое окно с последними записями журнала – просто щелкните на них для получения более детальной информации.

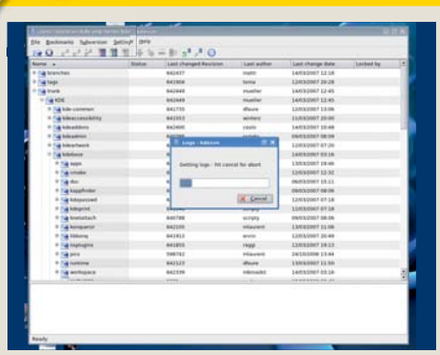
Другая полезная функция – возможность открытия любого отдельного файла в подходящем приложении щелчком правой кнопкой мыши и выбором привычного пункта Открыть с помощью... в стандартном меню. Но это далеко не все. Наряду с просмотром других SVN-репозиториев вы можете несколькими щелчками создать и администрировать свой собственный. Затем вы можете обновить локальную копию и продолжить работу над ней. Хотите зафиксировать кое-какие файлы? Просто перетащите их в соответствующее место дерева, и появившийся диалог запросит сообщение для журнала. Можно, естественно, перенести на главную панель и сразу несколько файлов. Мы нашли, что при работе с представлением в виде дерева механизм drag-n-drop может работать чуть-чуть неточно, но это, похоже, единственная часть приложения, нуждающаяся в исправлении. *Kdesvn* не может победить нищету в мире или сделать вас попривлекательнее, но он работает, и работает хорошо.



» *Kdesvn*: можете скачать версию по состоянию на любой момент времени в прошлом.

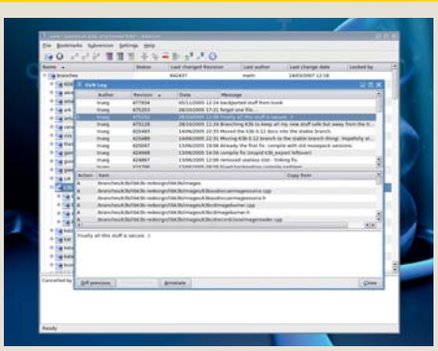
> «Такие функциональные инструменты сберегают нам время на удовольствия.»

Шаг за Шагом: Загрузка и выгрузка версий в SVN



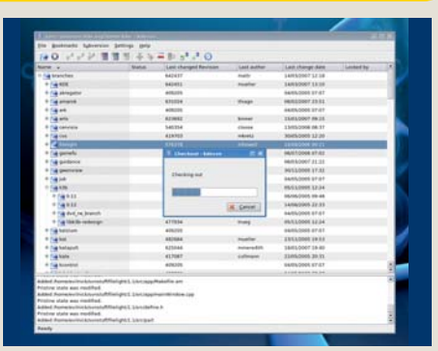
» Найдите ваш проект

Найдите желаемый репозиторий и введите URL (обычно svn://svn.something.org) или настройте свой собственный локальный проект!



» Загрузите копию проекта

Просмотрите журнал работ и выберите желаемую ветку кода или отдельный файл. При загрузке из репозитория будет создана локальная копия.



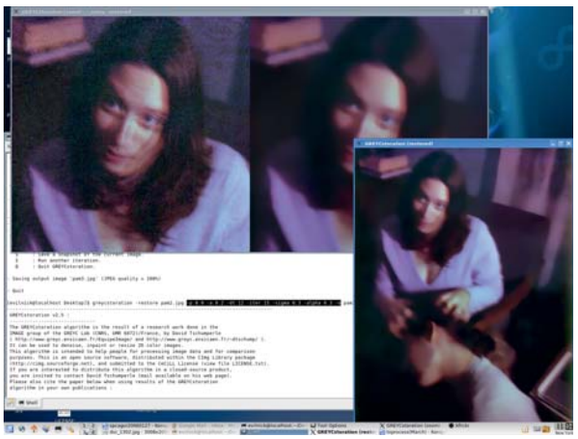
» Выгрузка изменений

Отредактируйте файл. Щелкните на кнопке Commit и введите сообщение для журнала работ, чтобы выгрузить ваши изменения обратно в репозиторий.

Графический редактор

GREYCstoration

Версия 2.5 **Сайт** http://snipurl.com/1ddo8



› **Пэм по жизни не такая уж симпатяга, но *GREYCstoration* творит чудеса!**

Общеизвестно, что если приложение имеет длинное неудобопроизносимое имя с якобы случайно расставленными заглавными буквами, значит, это командный интерфейс, который делает нечто конкретное. *GREYCstoration* – из этого разряда, и его конкретные действия весьма полезны.

Нет на свете совершенства: ни Майк, ни *Linux Format*, ни даже сам редактор не идеальны. И, конечно же, не идеален наш фотоаппарат: как и многие другие, он склонен добавлять лишние шумы даже при использовании наилучших установок ISO. Проприетарный мир полнится решениями для пост-обработки этой проблемы, но они дороги и не всегда успешны. Использование *GREYCstoration* не лишено трудностей, зато позволяет получить фантастические результаты.

Он использует уравнения в частных производных для анализа структуры изображения, и сглаживает или размывает пиксели в найденных направлениях. Главное, чтобы приложение умело определять, что является шумом, а что – структурой. Это у нас цветовой отблеск или деталь бархатного пиджака? Мир ИИ еще не решил эту проблему, а покамест мы можем поиграть с загадочными параметрами *GREYCstoration* и подыскать значения, которые сработают. Лучше всего начать со стандартных настроек и посмотреть, к чему это приведет:

```
GREYCstoration -restore noisy.jpg -p 0.7 -a
0.73 -dt 15 -iter 20 -sigma 0.5 -alpha 0.3 –o
clean.jpg
```

В интерактивном режиме вы можете нажать I (в активном окне) для получения следующей итерации. Истинное мастерство здесь – это умение определить, сравнивая исходное и получившееся изображения, что именно следует подправить для улучшения результата.

Хотя во многих случаях эксперименты с инструментом командной строки более эффективны, когда вы знаете, что делаете, есть также модуль расширения для Gimp, доступный по адресу http://registry.gimp.org/plugin?id=6233.

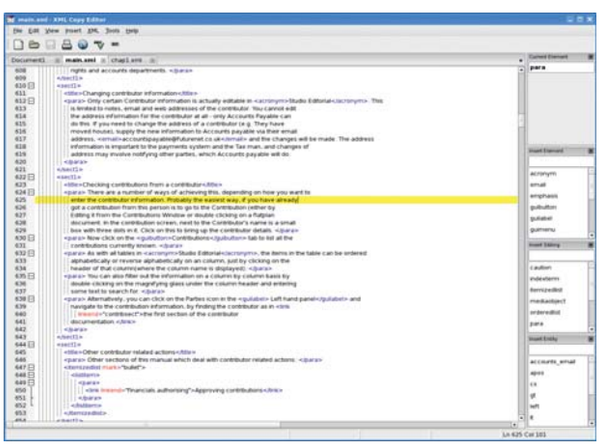
Вообще-то мы приврали, говоря, что он делает нечто конкретное, потому что кроме удаления шума из изображений, *GREYCstoration* выполняет также закрашивание (удаление из изображения некоторых особенностей и замещение их так, как будто ничего не было), интеллектуальное изменение размеров, используя технологии увеличения изображений без появления размытости или квадратов. Более подробно о режимах работы можно узнать на домашней странице!

> «Анализ изображений использует уравнения в частных производных.»

XML редактор

XML Copy Editor

Версия 1.0.9.1 **Сайт** http://xml-copy-editor.sourceforge.net



› **XCE лишен некоторых функций для опытных пользователей, но хорош для начинающих.**

Если в нашем журнале еще хоть раз в одном предложении встретятся акроним XML и выражение ‘lingua franca’, мы боимся, микросхемы в вашем мозгу перегорят. Но сейчас вы нас простите. Ведь нельзя отрицать, что мир IT принял прекрасные возможности расширяемого языка разметки и пытается применять его повсюду. Ура! Теперь все имеет контекст, структуру, атрибуты и валидность. Э-э… за исключением инструментов, используемых для создания XML с нуля.

Как ни странно, хороших инструментов для редактирования XML вовсе не так много. Да, можно просто настучать его в старом добром текстовом редакторе. Даже базовыми инструментами типа *Kate* можно попытаться обтесать XML в нечто пригодное для работы. Но, конечно же, достойный XML-редактор должен походить на IDE для XML и иметь автозавершение, сворачивание, валидацию и прочее. Он должен мгновенно перемалывать ваш XML и предоставлять DOM-инспекторы, проверку в реальном времени и другие классные штучки, до которых вы пока даже не додумались.

Не будем долго томить вас догадками. *XML Copy Editor* обладает прекрасными функциями: подсветкой синтаксиса, нумерацией, проверкой ошибок, форматированной печатью и многим другим. Он поддерживает DTD, XML и схему валидации RELAX NG.

Было бы здорово, если бы разработчики расширили инструментарий этого редактора, ради сложных операций вроде рефакторинга и пост-обработки XML с использованием чего-нибудь вроде *Jade*. В настоящее время, вероятно, лучший XML-редактор – это *Oxygen* (www.oxygenxml.com), но он стоит денег, работает под Java, да еще и проприетарный. Более того, он, вероятно, делает массу ненужных вам вещей.

> «Как ни странно, хороших XML-редакторов вовсе не так много.»

Пусть вы можете обойтись кодированием в *Vi*. Можно справиться с этой задачей и в *Emacs* – и, честно говоря, если вы выучили большую часть его клавиатурных комбинаций, вероятно, лучше в нем и остаться. Часто хорошее знание одного инструмента перекрывает преимущества от перехода на более специализированное. Но ведь в таком случае легко потерять что-то стоящее. Может, *XML Copy Editor* мир и не перевернет, но список его функций выглядит весьма впечатляюще, и даже сейчас это быстрая и удобная в использовании среда для создания XML.

Графический редактор

Raw Studio

Версия 0.5 **Сайт** www.rawstudio.org

Изобретение цифровых фотокамер принесло множество новых форматов файлов. Большинство фотокамер среднего и высокого качества используют сенсоры, захватывающие линейное 12-битное значение для каждого пикселя каждого цвета. Эти исходные данные принимаются зарядовыми или другими устройствами, и обычно обрабатываются в фотокамере для получения JPEG с 8-ю битами на пиксель. Так что четыре бита информации пропадают!

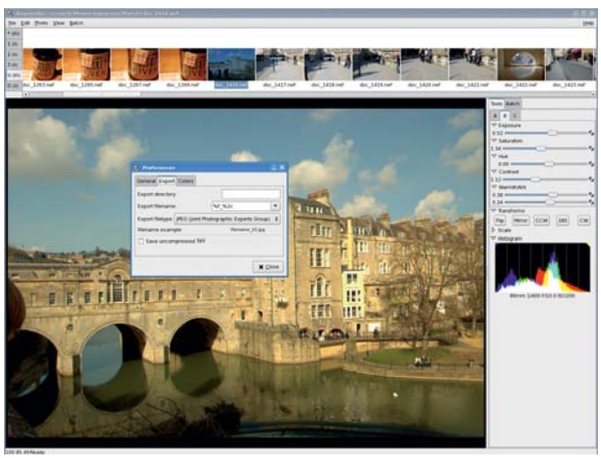
Не все так просто, потому что камера перед записью файла на карту памяти выполняет и другую обработку. Но одержимые фотографы гневаются: с чего это камере позволено решать, какие 8 бит взять из 12? По этой причине (плюс стимуляция продаж довольно дорогого программного обеспечения) многие изготовители камер встраивают возможность сохранения необработанных данных.

К сожалению, изготовители не пришли к согласию по способу записи этих данных – скажем, файлы необработанных данных от камер Canon абсолютно не похожи на файлы Nikon, Pentax или Olympus. Все эти форматы одинаково называются RAW-файлами, но по жизни пользоваться ими весьма сложно (иногда они меняются от камеры к камере – даже от одного изготовителя).

Сторонников полного контроля над данными при открытости кода выручает превосходная и стойкая подмога Дэйва Коффина [Dave Coffin] и его утилиты *Dcraw*, поддерживающей огромное и постоянно растущее число этих форматов (96 камер). Но хотя *Dcraw* – фантастическое средство манипуляции изображениями через командную строку, здесь отсутствует возможность немедленного просмотра результата. *Dcraw* используется как основа в других проектах, и на ней построено несколько расширений для *Gimp* и отдельных приложений; одним из лучших является *Raw Studio*, новая (0.5) версия которого только что вышла.

Всего лишь лучшая

Raw Studio не особенно проста в использовании, но зато обеспечивает полный контроль над результатом, включая поддержку управления цветовыми профилями (с *LittleCMS*); все предвидимые элементы настройки удобно сосредоточены справа от изображения. Три вкладки позволяют испробовать различные стратегии улучшения изображения, переключаясь между ними для выбора наилучшей. Изображениям можно также присвоить один из трех уровней приоритета, чтобы упростить выбор снимков из раздутого каталога «сырых» данных.



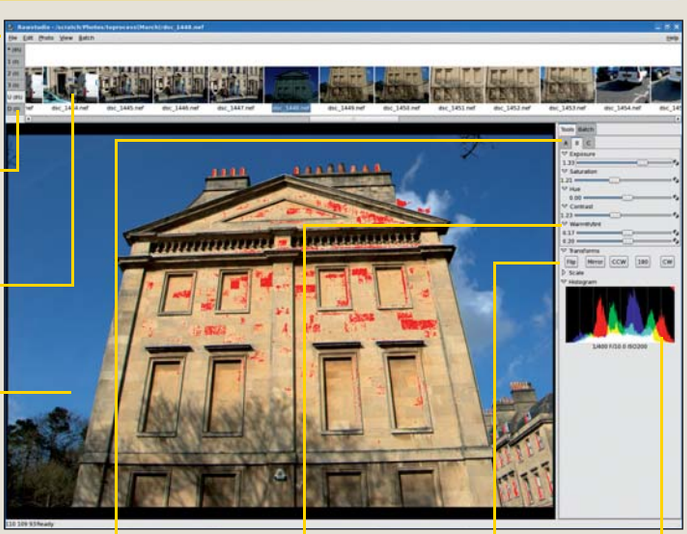
› ***Raw Studio*** **поможет выжать максимум из этой дорогой камеры.**

Просто укажите *Raw Studio* каталог с файлами, и панель предварительного просмотра заполнится RAW-файлами из каталога (а также PNG- и JPEG-файлами, если выбраны соответствующие опции). Пролистайте список и назначьте снимкам приоритеты. Затем вы сможете открыть соответствующую данному приоритету вкладку и просмотреть выбранные вами снимки, регулируя экспозицию и прочее. Да, чтоб не забыть: используйте маску экспозиции! Включите ее, нажав Ctrl+E, и «передержанные» области изображения (для тех, кто не одержим фотографией – снятые со слишком большой выдержкой) будут подсвечены красным. А области, где детали тонут в черноте, окрасятся в синий.

Новшеством в версии 0.5 является пакетная обработка. *Dcraw* настолько хорош (а во время работы использует расширенные возможности процессоров вроде MMX и SSE), что обработка изображений не отнимает много времени, но если вы тащитесь по большому списку изображений или имеете менее быстродействующее оборудование, то очень здорово поместить изображения в специальный пакет, а затем просто запустить пакетную обработку и заняться другими делами.

Raw Studio не идеален – для редактирования изображений здесь имеются только базовые функции коррекции и обрезки. Но зачастую это не столь важно: для сложных манипуляций с изображением *Gimp* все равно лучше. Если же вы хотите лишь выжать максимальное качество из вашей камеры, это прекрасный способ предварительно обработать файлы, а уж потом загнать их в *Gimp*.

Изучаем интерфейс Raw Studio



Умные параметры
Установите параметры для работы с изображениями и управления цветом.

Приоритеты
Организуйте файлы с различными уровнями

Миниатюры
Миниатюры текущих изображений.

Маска экспозиции
Главное окно изображения. Нажмите F11 для полноэкранного режима или Ctrl+E для маски экспозиции.

Переключайте вкладки
Три вкладки означают, что можно экспериментировать с настройками.

Подстройка
Отредактируйте изображение при помощи ползунков.

Преобразование
Обрезка и масштабирование.

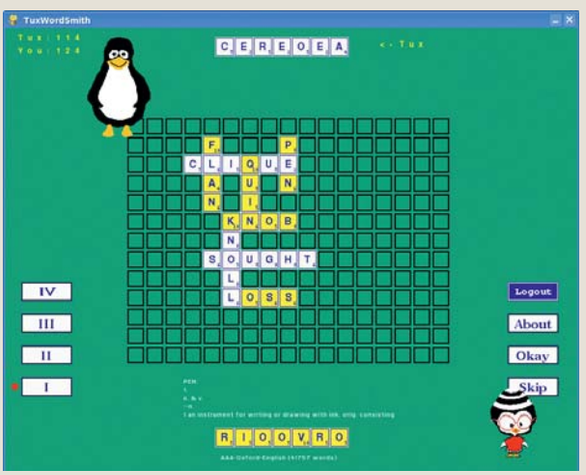
Гистограмма
Отображает текущую гистограмму.

HotGames Развлекательные приложения

Словесная забава

TuxWordSmith

Версия 0.3c **Сайт** www.asymptopia.org



› Пока вы пишете без ошибок, V-I-C-T-O-R-Y будет за вами.

Знаете, ewt – очень полезное слово. Очень, очень полезное при игре в слова, хотя, может, и не так, как em и en, которые часто можно нарастить сразу в двух направлениях и заработать очки, если вы играете в «Эрудит» или нечто подобное. В нашем случае, нечто подобное – это игра *TuxWordSmith*, вовсе не похожая на «Эрудит», для юридического отдела Hasbro будь сказано. Разве что вы тоже получаете очки за размещение на доске слов, создаваемых из отдельных букв-плиток.

По умолчанию *TuxWordSmith* выставляет против вас Тукса (ластоногого компьютерного игрока) для словесной дуэли на английском языке или на любом из поддерживаемых (испанский, немецкий и даже латынь). Очки за каждую букву-плитку начисляются динамически в зависимости от частоты их появления в соответствующем словаре. Это может слегка сбить вас с толку, если вы привыкли к другой системе начисления очков в играх с буквами, потому что английская “B”, например, оценена в девять очков вместо трех. (Вам следует пересмотреть свой словарь полезных коротких слов.)

Игровой процесс весьма медлителен, графика устаревшая и грубая, и некоторые слова без видимых причин отсутствуют в словаре. Мы не хотим сказать, что Тукс жульничает: он и не в силах это сделать, так как вам видны все буквы у него на руках. Но иногда он долго чешет в затылке и вытягивает из глубин своего индексированного массива нечто обескураживающее.

В любом случае *TuxWordSmith* скорее предназначен для образовательного рынка. По этой части он имеет индивидуальные учетные записи для игроков, так что можно вести таблицу результатов, определять словари и прочее. Честно говоря, этот аспект игры выглядит не в меру усложненным; и поскольку у нас еще нет класса, заполненного студентами-латинистами, мы просто играли за админа, игнорируя туманные настройки особенностей игрока. Но наши умные читатели, вероятно, смогут с этим справиться.

> «Вас выставляют против Тукса (киберигрока) для словесной дуэли.»

Пошаговая стратегия

UFO: AI

Версия 2.1.1 **Сайт** http://ufo.myexp.de

› Как вы знаете, пришельцы пользуются Mac’ами.

Странные вещи творятся с рогатым скотом планеты Земля. Его злобно губят, затем расчленяют, поджаривают и запихивают в булку – получается завтрак для Майка. В других местах бывают вещи и похуже: мерзкие пришельцы валятся с небес в своих космических кораблях-блюдцах, похищая людей и используя небоскребы и лачуги обитаемых континентов как учебные цели для стрельбы из оружия иных миров.

Да, мы играем в *UFO: Alien Invasion*. Если вы никогда не играли в *UFO: Enemy Unkown*, вам следует немедленно найти копию на eBay (кстати, DOS-версия весьма похожа на Amiga-версию). Если играли, то вы знаете чего ждать: пошаговое тактическое управление командой в различной обстановке, упакованное в стратегическую игру со своим сюжетом, управлением ресурсами и исследованиями.

Но кое-что отличается. После нескольких лет игры в *UFO: EU* и ее продолжения, вы готовы поклясться, что узнаете, куда двинутся пришельцы, по сверканию пикселей в их глазах (или другой оптике). *UFO: AI* меняет правила, используя правдоподобную графику! Основанная на движке *Quake*, пошаговая битва теперь происходит в прекрасно оформленной 3D-среде. Нет больше ограничений на изменение вида: вы легко можете поворачиваться, перемещать точку обзора и изменять масштаб.

Надо признаться, что игра все еще не доделана. Общие стратегические компоненты нуждаются в доработке (например, базовый менеджмент, наем и увольнение и т.д.). Известно также, что игра норовит рухнуть, даже когда вы пытаетесь выполнить относительно простые вещи. Но сами миссии работают отлично. Более того: превосходно. Постоянно кажется, что пришельцы выскакивают прямо перед вами, хронически не хватает единиц времени, чтобы бросить гранату с уже выдернутой чекой, и бедная корова Дэйзи, как всегда, обречена. Обязательно искупайтесь в теплой ванне переработанной цифровой ностальгии.

> «Миссии работают отлично. Даже не отлично, а превосходно.»

Антивирус

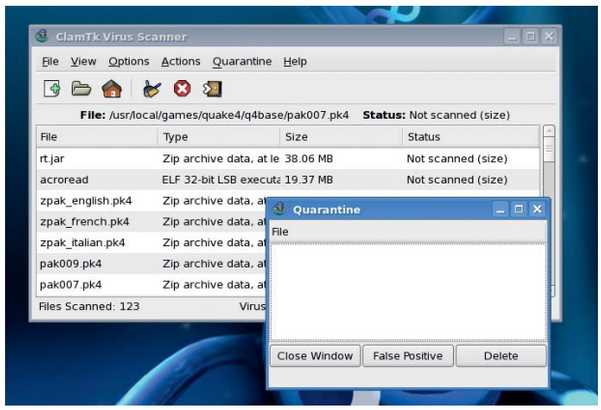
ClamTk

Версия 2.3 **Сайт** http://clamtk.sourceforge.net

Ничего привлекательного в проверке на вирусы нет. Она дико скучна. Скучнее, чем объяснения тонких мест грамматики нашими корректорами. Правда! Но от нее никуда не денешься. Мы в этом плане, вообще-то, не маньяки, но есть немало доводов в пользу поиска вирусов в файлах, особенно если вы, как большинство людей, храните файлы Windows на Linux-сервере. Среди всего этого хлама из Сети, в одном-двух файлах наверняка затаился неприятный сюрприз.

Самый известный сканер вирусов с открытым кодом – *ClamAV*. Часто обновляясь, вы можете быть вполне уверены, что он обнаружит большинство вирусов и троянов до того, как они успеют нагадить. Но, прекрасно работая в качестве демона и осуществляя регулярное сканирование, как ему велено, этот прекрасный движок не имеет графического интерфейса, удобного для обычного пользователя. Так что вполне понятно, где именно за дело берется *ClamTk*.

Интерфейс *ClamTk* слегка упрощает запуск внепланового сканирования вашей свежескачанной добычи и позволяет обрести уверенность в том, что **злобныйвирус.exe** – файл, в действительности, непорочный. GUI (который, как вы догадываетесь, основан на GTK) вполне понятен и позволяет сканировать файлы в каталоге, каталоги и подкаталоги, или отдельные файлы. Подозрительные файлы (или те, что в данной конфигурации пропускаются) отображаются на главной панели. Подозрительные элементы могут быть посажены в карантин или просто удалены. Текущий список карантина управляется одноименным меню; его элементы можно просеивать и помечать для восстановления или удаления. Забавно, что антивирусное ПО почти всегда попадает под подозрение, потому что содержит описания вирусов – мы хлебнули горя, пропихивая сам *ClamAV* через нашу систему дублирования дисков!



Вот подхватим реальный вирус, и тогда покажем, на что способен *ClamAV*!

> «Запуск внепланового сканирования сомнительных закачек.»

Хотя вы, вероятно, регулярно обновляете файлы сигнатур, если запускаете демон *ClamAV* (файлы сигнатур – это такая штука, помогающая приложению обнаружить вирусы), обновление можно выполнить и вручную из меню Help. Последнее, что стоит отметить – компонент *ClamTk-kde*, связанный с *Konqueror*: благодаря ему для проверки любого файла вы можете просто выполнить на нем правый щелчок в браузере *Konqueror* и выбрать Действия > Сканировать на вирусы. Тут откроется полный интерфейс *ClamTk*; без сомнения, это довольно удобно, если вы мнительны и подозрительны.

Монитор сети

Nagios

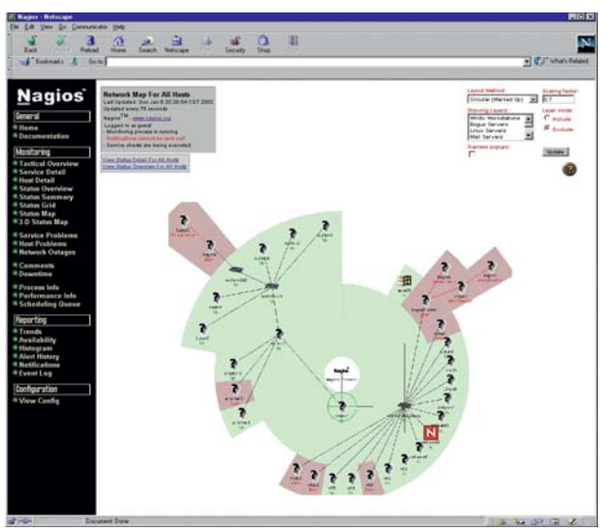
Версия 3.0a **Сайт** www.nagios.org

LINUX FORMAT HotPicks повторный визит

Чтобы ваша Сеть (Net) всегда была в состоянии «Есть» (а не «Нет»), вы должны приглядывать за вашими серверами. Да ведь следить за ними ужасно нудно, скажете вы; нельзя ли поручить слежение кому-нибудь другому? В *Linux Format* для этого есть должность ‘Майка’, но наше решение – не панацея. Практичнее будет установить какое-либо приложение мониторинга. Тогда, правда, придется приглядывать уже за приложением, но, по крайней мере, только за ним – глаз-то у вас всего два.

Nagios уже не раз рассматривался в *Linux Format* и за последние годы превратился в широко используемый и уважаемый инструмент мониторинга сети. Настройте систему правильно, и web-интерфейс на базе CGI предоставит вам быстрый обзор ваших серверов и всех текущих проблем. Можно контролировать широкий диапазон служб, включая стандартные вещи вроде доступности конкретной службы (SSH, HTTP и так далее), и состояние системы: загруженность, свободное пространство и прочее. С некоторыми браузерами *Nagios* может даже издавать звуки при наступлении определенных событий, например, звук ‘пинг’ при проблемах с *ping*. Согласитесь: созерцать красивые графики и даже 3D-отображение состояния вашего сервера куда веселее, чем пялиться на ‘Майка’.

Не стоит даже мечтать об установке подобной системы одним щелчком. *Nagios* требует, чтобы вы добавили в систему пользователя ‘*Nagios*’ и перенастроили *Apache* на машине, которая будет осуществлять мониторинг, указав на установленные файлы web-интерфейса *Nagios*. Придется также посидеть с документацией, чтобы разобраться, как правильно настроить систему, и убедиться, что вы следите за тем, чем надо. К счастью, отменная документация к стабильному релизу выручит вас при любых экспериментах со свежим билдом от разработчиков – но, пожалуйста, не доверяйте недоделанному приложению работу с критичными для вашего бизнеса серверами, от которых требуется надежность на уровне «пяти девяток» (99,999%). Оно покамест просто для забавы!



Ваши серверы должны отображаться на диаграмме примерно вот так!

> «Можно заставить его издать звук ‘пинг’ при проблемах с ping.»

Мы, разумеется, опубликуем полный учебник по *Nagios 3.0* в будущих номерах, когда выйдет окончательная версия. До этого – обойдитесь без кошмаров...

Монитор активности

KeyFrog

Версия 0.8 **Сайт** www.keyfrog.org

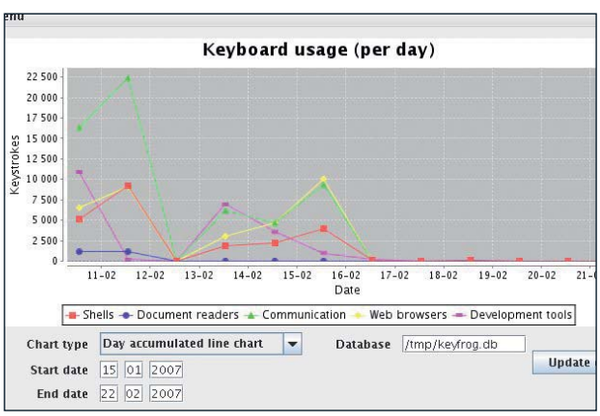
Бывали ли у вас дни, когда вы недоумевали: а чего же я за сегодня достиг? Если речь идет всего лишь о днях, то вы счастливчик; у нас, например, случались целые недели, за которые буквально нечего было вывесить на наш мысленный стенд достижений. А может, вы, наоборот, из тех, что поспевают сделать три дела еще до завтрака. В любом случае, неплохо знать, куда на самом деле девается время.

Linux, естественно, ведет учет процессов, так что вы может отследить, какие приложения запускались на вашей машине. Но отсюда не ясно, какое приложение активно использовалось. *KeyFrog* – это попытка предоставить более информативную статистику: отслеживая нажатия клавиш и события на экране, сообщить, что и когда вы делали.

Простой файл настроек позволяет указывать приложения и объединять их в группы – например, определить *Ekiga*, *Thunderbird* и *Pan* как группу 'общение'. Затем *KeyFrog* будет записывать 15-минутные обновления действий в файл базы данных *SQLite*, который впоследствии можно обработать приложением *KeyVis*, поставляемым с исходными текстами.

Работает *KeyFrog* прекрасно, но утомительной процедуре сборки *Kvis* буквально нет конца – лучшим советом, вероятно, будет использовать *NetBeans*. Так что этот пакет нуждается в доработке, чтобы стать подлинно полезным. Но по нашему мнению, труды будут вложены не зря.

› Если вы сможете заставить работать *Keyvis*, то даже получите красивые графики!



Менеджер ISO-образов

AcetoneISO

Версия 6.7 **Сайт** www.acetoneteam.org

AcetoneISO (подсказка в имени) – это инструмент для работы с ISO-образами: ну, большими такими файлами, являющимися образами CD или DVD. Более конкретно, это инструмент для монтирования образов, преобразования между форматами и вообще взаимодействия с такими файлами.

В последнюю версию добавлена поддержка DAA, BIN, NRG и MDF (про последний я всегда думал, что это продукт с лесопилки, но вообще-то это родной формат для Windows-приложения *Alcohol*). Эти форматы поддерживаются только в том случае, если содержат ISO-образы, а не их сложные или загадочные версии. Вследствие полного отсутствия защиты от копирования на Linux-дисках, вы можете удивиться, кому нужна поддержка (а она есть) таких форматов, как CUE и NRG. Суть в том, что эти форматы широко используются 'другими' программами на 'различных' платформах, так почему бы не предусмотреть работу с ними и в Linux. Полезно также иметь возможность открыть ISO-файл DVD прямо в *MPlayer* или где-то, не монтируя файл в командной строке как loop-устройство.

Интерфейс действительно прост. Он весь состоит из серии простых диалогов (которым не помешало бы чуть большее внимание к графике) для монтирования, проигрывания, конвертирования и прожига образов дисков. Если у вас установлено приложение *P7Zip*, то вы также можете сжимать их до приемлемых размеров. LXF

› Монтировать, сжимать, прожигать, риповать и проигрывать ISO-образы легко.

Также выпущены

Новые и обновленные программы, заслуживающие вашего внимания...

Bacula 2.0.3. Резервирование на ленту и диск через сеть www.bacula.org

Dynebolic 2.4.1. Live-дистрибутив с мультимедийными инструментами www.dynebolic.org

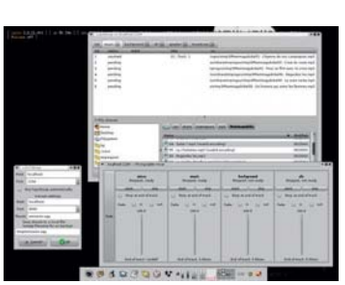
› Dynebolic: Живая медиа-студия.

GNU Parted 1.8.3. Уверенное редактирование разделов диска www.gnu.org/software/parted

Leafnode 1.11.6. Небольшой сервер новостей www.leafnode.org

LinuxBIOS 2-2570. Холодный старт Linux с вашей материнской платы. www.linuxbios.org/index.php/Main_Page

Liquidsoap 0.3.2. Обработка и потоковое воспроизведение звука http://savonet.sourceforge.net/wiki/Liquidsoap



› Создайте аудио-приложение с Liquidsoap.

Lives 0.9.8.3. Система проигрывания видеоэффектов/преобразований http://lives.sourceforge.net

NoMachine NX 2.1.0-5. Клиент/сервер удаленного рабочего стола на базе X11. www.nomachine.com

Pango 1.16.0. Необходимые библиотеки для интернационализации www.pango.org

Recovery Is Possible 2.2. Загрузочный спасательный CD/DVD www.tux.org/pub/people/kent-robotti/looplinux/rip

Sylpheed 2.4.0-beta5. Приятный легковесный почтовый клиент http://sylpheed.sraoss.jp/en

Wine 0.9.33. Запускайте MS-приложения без MS www.winehq.org

LXF DVD92

Изголодались по новым операционкам? Ну, держитесь…

Майк Сондерс любовно подбирает содержимое диска Linux Format, а также поддерживает сайт www.linuxformat.co.uk.

Чем больше, тем веселее

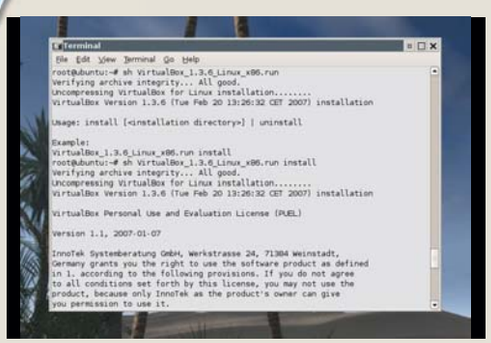
Благодаря обратной связи от вас мы в курсе, что вы любите читать в *Linux Format* об альтернативных операционных системах. Наши обзоры FreeBSD, Syllable и OpenSolaris привлекли всеобщее внимание, и это отлично, что вам нравится изучать и другие ОС, а не только Linux. Однако в огромном и скверном Интернете дела обстоят вовсе не так мирно: ежедневно я натыкаюсь на доморощенных знатоков операционных систем, которые предают анафеме разработчиков, на беду не разделяющих их личных пристрастий. У некоторых прямо кровь вскипает при мысли о том, что кодеры трудятся над чем-то кроме Linux, *OpenOffice.org* и *Firefox*.

Все эти ораторы полагают, что Open Source не видать успеха, пока все мы не сольёмся в одном проекте. Они наивно верят: вот утопим проект FreeBSD, и все разработчики тут же переключатся на Linux и займутся настоящим делом. Но дело-то в том, что разработчики открытого ПО программируют исключительно из любви к искусству – им дороги их проекты, и они не станут перепрыгивать с одного на другой.

И здесь есть еще один важный момент. Сейчас мы подхихикиваем над теми, кто беснуется в зале суда SCO, но пару лет назад вполне серьезно опасались, что использование Linux может стать незаконным – по крайней мере, пока не будут удалены некоторые части кода. Но поскольку разработка Open Source не сошлась клином на Linux, у нас еще оставалась FreeBSD, на которую мы бы положились (и продвигали), если бы на решение юридических проблем Linux понадобились годы. Ну, а если по какой-то причине и BSD увязнет в судебных разбирательствах, у нас есть Haiku, Syllable, ReactOS и еще много чего, чтобы использовать и усовершенствовать.

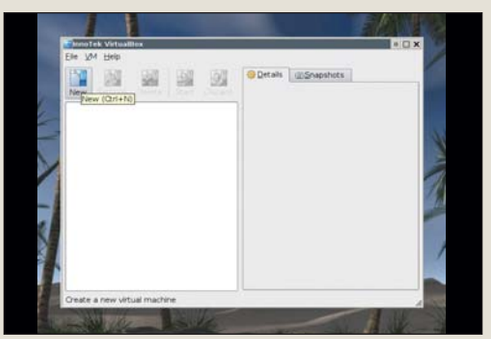
Монокультура ослабляет; так давайте радоваться жизнеспособности и многообразию рабочих открытых ОС!

Шаг за шагом: Запускаем ОС в VirtualBox



1 Установка

Скопируйте **VirtualBox_1.3.6_Linux_x86.run** в домашнюю директорию. Откройте терминал, переключитесь в режим администратора (**su** или **sudo bash**) и введите **sh VirtualBox_1.3.6_Linux_x86.run install**.

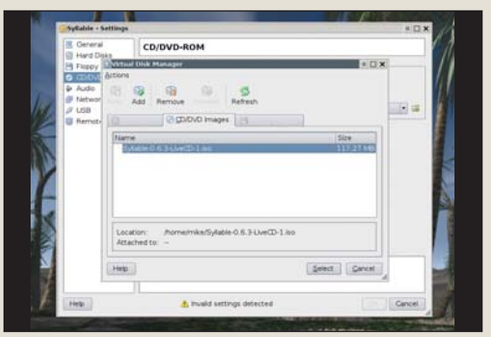


2 Пуск

Добавьте пользовательскую учетную запись в группу ‘vboxusers’, введя **usermod -G vboxusers <username>** от имени администратора, затем от имени обычного пользователя запустите **/opt/VirtualBox-1.3.6/VBox.sh**.

3 Создание

Нажмите New, чтобы запустить создание виртуального ПК. Выберите тип ОС – Other/Unknown, выберите 128 МБ ОЗУ и создайте жесткий диск с фиксированным объемом 2 ГБ.



4 Запуск

Тут перед вами появится этот экран. Нажмите CD/DVDROM, Mount и ISO Image File; выделите образ CD, который вам нужен на нашем DVD. Затем – OK и Start, и процесс пошел!

Операционные системы

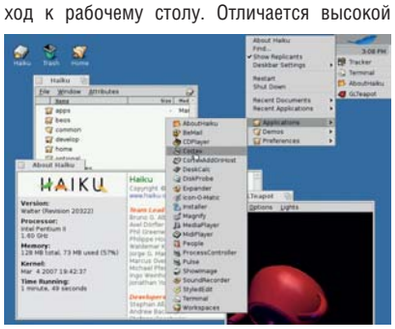
Мегапак альтернативных ОС

В этом месяце мы собрали для вас, на пробу, коллекцию новых интересных ОС, представленных в виде образов CD и жесткого диска, так что вы сможете запустить их с помощью эмуляторов ПК, которые тоже включены на DVD – следуйте инструкциям, приведенным ниже, чтобы запустить *VirtualBox* и *Qemu* с операционными системами из раздела **Дистрибутивы**. Поскольку *Qemu* отличается лучшей совместимостью, мы дадим вам несколько дополнительных подсказок по работе с *Qemu*.

Syllable (**www.syllable.org**), которая рассматривалась в LXF78, применяет новый подход к рабочему столу. Отличается высокой скоростью и интегрирует в ОС графический пользовательский интерфейс – то есть вам не придется работать с миллионом оконных менеджеров и наборов инструментов. Мы включили в содержимое диска Live CD образ; скопируйте его на жесткий диск, распакуйте, затем запустите в *Qemu* с помощью:

```
qemu -cdrom Syllable-0.6.3-LiveCD-1.iso -
    boot d -m 128
```

Войдите в систему как администратор с паролем ‘root’.



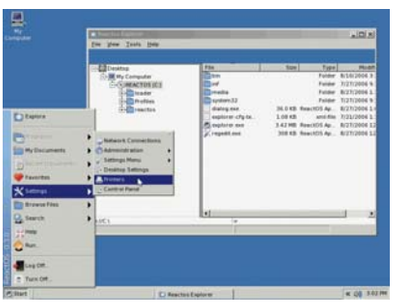
› **Haiku, быстрый и дружелюбный к пользователю клон BeOS, годится Linux в союзники.**

ReactOS (**www.reactos.org**) – это клон Windows с открытым кодом. Он пока на начальной стадии разработки, но уже способен запускать ранние версии *MS Office* и *Unreal Tournament*. На нашем DVD вы найдете Live CD и образы установочных CD – распакуйте их и загрузите в *Qemu*. Например, чтобы открыть Live CD, установите *Qemu* в соответствии с нижеприведенным описанием, скопируйте и распакуйте файл **reactos-0.3.0-live.zip** в домашнюю директорию и введите:

```
qemu -cdrom reactos-0.3.0-live.iso -
boot d -m 128
```

Еще один проект ОС-клона – Haiku (**www.haiku-os.org**), стремящийся наделить суперскоростной рабочий стол BeOS очарованием Open Source. Скопируйте **haikuimage-070307.bz2** из **Distros/Haiku** в домашнюю директорию и введите:

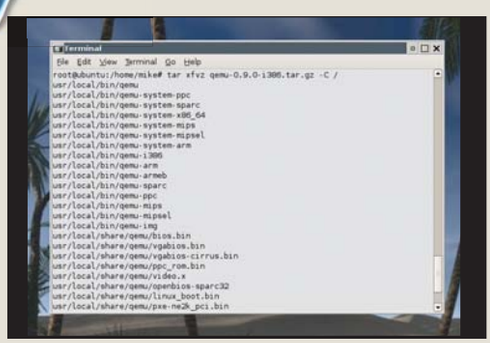
```
bunzip2 haiku-image-070307.bz2
qemu -hda haiku-image-070307 -m 128
```



› **Стиву Балмеру [Steve Ballmer] придется ухватиться за стул – ReactOS, клон Windows с открытым кодом.**

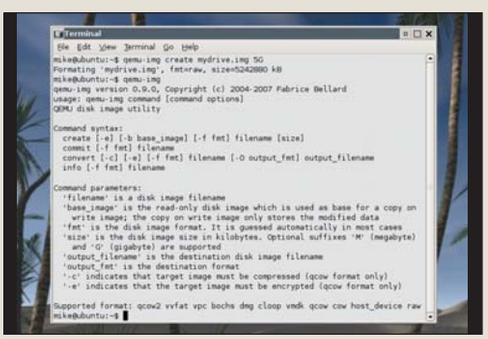
Вам мало? FreeDOS (**www.freedos.org**) – клон DOS с открытым кодом, умеющий работать в режиме Live; испытайте его с вашими старыми DOS-приложениями. Minix 3 (**www.minix3.org**) – самая новая версия ОС, некогда вдохновившей Линуса Торвальдса на создание Linux; введите **qemu -hda minix-3.1.1.img**, чтобы запустить ее, и войдите в качестве администратора (без пароля). Если вы – продвинутый пользователь Linux, попробуйте собрать ускоритель *KQemu* (см. **kqemu-1.3.0pre11.tar.gz** в разделе **System/Qemu**) и добиться космических скоростей. См. также стр. 109, там находится руководство по работе с BSD. »

Шаг за шагом: Запускаем ОС в Qemu



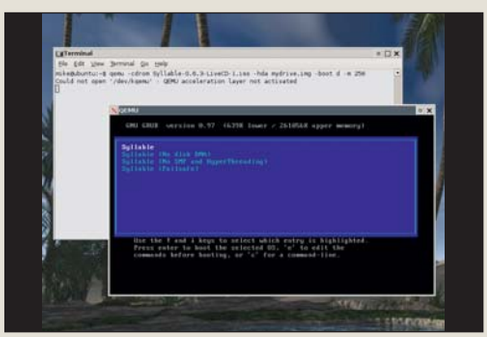
1 Установка Qemu

Для установки *Qemu* скопируйте **qemu-0.9.0-i386.tar.gz** из раздела **Система/Qemu** нашего DVD в домашнюю директорию и от имени администратора введите **tar xfvz qemu-0.9.0-i386.tar.gz -C /**.



2 Жесткий диск

Создайте образ виртуального жесткого диска (файл образа диска), введя **qemu-img create mydrive.img 5G**. Вместо **5G** можете указать любое количество гигабайт, которое вам нужно.



3 Запуск

Запустите *Qemu*, отметив образ жесткого диска флажком **-hda**, ISO-образ – **-cdrom**, порядок загрузки – **-d**, а RAM – **-m**, т.е., например, qemu **-cdrom disc.iso -hda mydrive.img -boot d -m 256**.

4 Остановка

Если вы кликнете где-то внутри окна, *Qemu* будет захватывать движения мыши. Нажмите Ctrl+Alt, чтобы вернуть настоящий указатель мыши на свое место.

Дистрибутивы Linux

Elive

Debian частенько высмеивают за нарушения намеченного графика, за упертость, за погрязание в политике, однако к его стабильности не придерешься. Как следствие, Debian лег в основу множества «более крутых» дистрибутивов, вроде Knoppix, Ubuntu, а теперь и Elive. По части названия приз за оригинальность ему не светит – начальная буква E из слова *Enlightenment* [просвещение, – прим. пер.] плюс поставка на Live CD – но зато он великолепен.

Enlightenment не особенно засвечен среди публики, но это оригинальный и красивый оконный менеджер. Спустя десятилетие разработки он дошел лишь до версии 0.17, но не переживайте по поводу номера: *Enlightenment* – солидный продукт, и многие используют его ежедневно. Внешняя привлекательность E предусматривалась для обеспечения вашего комфорта за рабочим столом, а вовсе не ради восторженных восклицаний, как у *Compiz* и Beryl.

Мы уже включали *Enlightenment* в наш DVD, но поскольку он опирается на целую кучу библиотек, работать с ним нелегко. И мы рады, что можем предложить вам испробовать *Enlightenment* без всякой компиляции и прочей головной боли: просто вставьте DVD в привод, перегрузите компьютер – и наслаждайтесь.

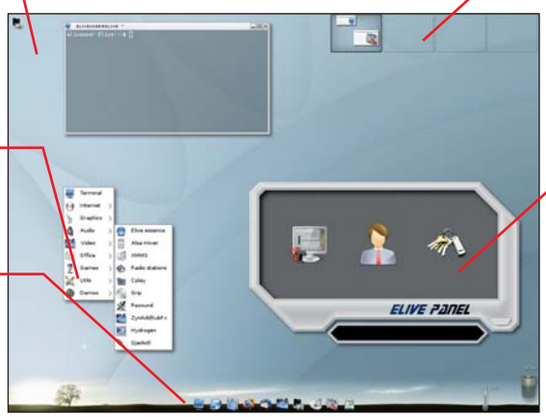
› Долой серость! Даже инструменты настройки выглядят в *Enlightenment* мило.

При загрузке Elive предложит вам несколько вопросов, чтобы определить вашу видеокарту, и несколько тем на выбор – берите тему по умолчанию. Вас также попросят установить разрешение экрана; выбирайте то же, что и при работе со своей ОС. Когда появится окно входа в систему, введите **eliveuser** как имя пользователя и пароль **elive** (кстати, пароль администратора также elive).

И тогда *Enlightenment* запустится! Внизу экрана вы увидите панель, подобную Mac OS X; поводите около нее курсором мыши, и значки будут пульсировать. Если вы кликните правой кнопкой по фону, появится меню имеющихся в Elive программ, а щелчок левой кнопкой откроет меню настройки *Enlightenment*. Обратите внимание, что окна в уменьшенном виде размещаются в виде значков на рабочем столе, и вы можете настраивать панель и переключатель рабочего пространства, щелкнув по ним правой кнопкой. Запустите программы, умилитесь на блестящие эффекты панели имен и всплывающих теней – и поймете, почему мягкие оттенки и приятный дизайн E завоевали ему армию преданных поклонников.

Elive – удобный портативный рабочий стол для Linux, с *Gimp*, *Firefox*, *Gaim*, *AbiWord*, играми и уймой других программ. Можно даже установить его на жесткий диск – но пока мы не советуем этого делать: программа установки еще в стадии разработки.

Исследуем рабочий стол Elive



Рабочий стол
Когда вы минимизируете запущенные программы, они появляются в виде значков на рабочем столе.

Меню
Щелчок правой кнопкой мыши на рабочем столе вызовет меню программ.

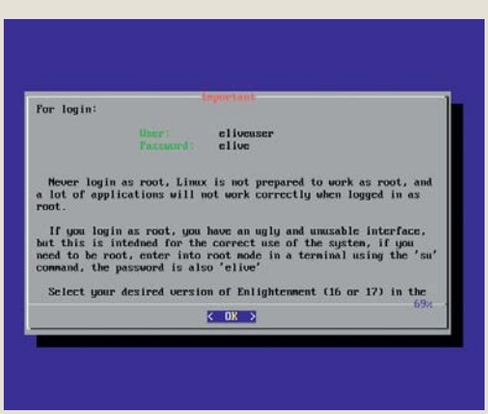
Док
Эта панель обеспечивает быстрый доступ к приложениям. Чтобы настроить ее, щелкните правой кнопкой мыши.

Настройка
Панель настройки Elive позволяет настраивать темы, шрифты и прочие косметические аспекты менеджера окон.

Рабочие пространства
Кнопки, позволяющие переключаться между четырьмя виртуальными рабочими столами. Можете даже перетаскивать значки с одного рабочего стола на другой.

Шаг за шагом: Запускаем Elive

1 Загрузка
Вставьте LXFDVD и перегрузите компьютер. При появлении этого экрана загрузки нажмите Enter или попробуйте 'failsafe', если при запуске Elive возникнут проблемы.



2 Пользование
В экране входа в систему введите в качестве имени пользователя 'eliveuser', затем – 'elive' в качестве пароля. Появится рабочий стол *Enlightenment*.

Операционные системы

BSD-клоны

Хотя технически они являются частью нашего мегапакета альтернативных ОС, FreeBSD, OpenBSD и NetBSD вполне заслуживают отдельного раздела. Постоянные читатели *Linux Format* знают, что мы уже не раз говорили о FreeBSD (**www.freebsd.org**) в предыдущих номерах, больше всего в LXF77, где рассказывалось о поддержке и о сообществе, создавшем такую суперсолидную ОС. Если вы никогда прежде не слышали о FreeBSD, вот вам краткая информация: подобно Linux, FreeBSD является Unix-подобной ОС с открытым кодом, разработанной для настольных компьютеров и рабочих станций.

В отличие от Linux, все компоненты FreeBSD разрабатывались в унисон, благодаря чему появилась очень цельная и четко спроектированная ОС – в противоположность Linux, где разные компоненты, например, ядро и *Glibc*, пишутся отдельно; в результате местами получается сборная солянка. Как следствие, у FreeBSD есть только один «дистрибутив», и эта ясность завоевала ей множество поклонников на сервере и в мире ISP. Мы рекомендуем всем линуксоидам попробовать FreeBSD; она не только расширит ваши знания о Unix, в ней – к счастью – работает 99% программ с открытым кодом для Linux, так что это – совсем не чужая пользовательская среда.

NetBSD (**www.netbsd.org**) и OpenBSD (**www.netbsd.org**) также являются Unix-подобными ОС с открытым кодом, причем первая особое внимание уделяет переносимости, работая более чем на 50 аппаратных платформах, а последняя известна своей нерушимой системой безопасности. NetBSD и OpenBSD не поддерживают обычного оборудования ПК в той же мере, что FreeBSD, поскольку чаще используются на серверах, чем на персональных компьютерах. NetBSD знаменита своей кодовой базой, одной из самых ясных и понятных среди операционных систем, и это идеальная учебная ОС для будущих проектировщиков.

В нашем разделе **Дистрибутивы*** вашему вниманию предлагается DVD-образ FreeBSD и CD-образы OpenBSD и NetBSD (x86 и AMD64). Можете запустить их с помощью *Qemu* или *VirtualBox*, как описано на предыдущих страницах, или записать образы на DVD/CDR и запустить на настоящем компьютере без эмулятора. Почитайте файлы TXT в директориях каждой ОС, там вы найдете руководство по установке и сведения о поддержке оборудования. Вот команды запуска FreeBSD, после копирования в домашнюю директорию файла **6.2-RELEASE-i386-dvd.iso**:

```
Qemu-img create freebsd-hd.img 5G
Qemu -cdrom 6.2-RELEASE-i386-dvd.iso -
    hda freebsd-hd.img -boot d -m 128
```

Они создадут виртуальный образ жесткого диска размером 5 ГБ и инициируют процедуру установки в *Qemu*.

› NetBSD работает везде, от солидных рабочих станций до, хм, тостеров!

3 Выход

Закончив работу, щелкните левой кнопкой мыши по рабочему столу, и выберите Exit Enlightenment'. Затем нажмите кнопку Power Off (внизу справа) в экране приглашения.

› Вы не расстанетесь с любимыми приложениями, работая во FreeBSD: *Firefox*, KDE, Gnome, *OpenOffice.org* и т.д. – все они здесь есть.

Документация

Журнал в формате PDF

Дочитав этот номер, можете обратиться к диску, там есть еще 50 страниц руководств, отобранных из предыдущих номеров. В этом месяце у нас руководства из раздела **Первые Шаги** (First Steps), предназначенные для новичков в Linux – но даже если вы уже не совсем зелёный, они все равно могут пригодится. Откройте **index.html** на DVD и найдите раздел **Журнал/PDFs** с перечнем статей.

Мы начали с трёхчастной серии статей про редактирование цифрового видео с помощью *Kino*. Это руководство поможет желающим подправить свое домашнее видео под Linux – например, добавить саундтрек или титры. Часть 1 объясняет, как импортировать домашнее видео в Kino, Часть 2 в деталях описывает процесс редактирования, а Часть 3 показывает, как записать вашу работу на DVD, чтобы дарить родным и близким.

По части администрирования системы у нас есть руководство из LXF68 по использованию *Apt* и RPM – весьма полезное чтиво, если система пакетов в Linux вас запутала – а также руководство из LXF71 по использованию инструментов сжатия. Если ваша домашняя директория забита мусором, а на рабочем столе полно папок под названием **разное** и **всякая всячина**, читайте мастер-класс LXF74 по очистке файловой системы.

› Выжмите из *Firefox* важнейшие сведения о безопасности с помощью нашего руководства.

И, наконец, предлагаем трио руководств по работе в Интернете. В разделе Первые Шаги LXF73 мы показываем, как работать со Skype, Google Talk и IRC, используя *Gaim*, а в руководстве из LXF78 открываем перед вами мир *Evolution*, почтового клиента Gnome по умолчанию. Если вы помешаны на проблемах безопасности, вам стоит познакомиться с нашим руководством из LXF81, объясняющим, как улучшить систему безопасности *Firefox*.

* Да, мы знаем, что технически все эти ОС не являются «дистрибутивами». Первый, кто напишет нам об этом, получит бесплатный флейм от Тео де Раадта.

Прочие приложения

Новые релизы

На DVD этого месяца не только одни ОС: там есть еще россыпи новых программ с открытым кодом. Для начала – *MonoDevelop 0.13*, основное обновление для популярной среды разработки C#. Если вы следили за нашими уроками по Mono, то вам стоит взять код с нашего DVD (см. раздел **Разработка**) и попрактиковаться с ним. Новые функции включают родную поддержку проектов *Microsoft Visual Studio 2005* и возможность определять настраиваемые команды для ваших проектов; он может даже создавать «типовые» проекты для других языков (например, для C).

Далее, в разделе **Рабочий стол**, имеется *Pinot*, инструмент «личного поиска и метапоиска», подобный *Beagle*. Традиционные инструменты поиска выполняют поиск только по именам файлов, а *Pinot* ещё и вникает в их содержимое. То есть если вы велите *Pinot* поискать «Зеленка», он не только будет искать имена файлов, содержащие строковую переменную «зеленка», но и просматривать документы с другими именами, содержащие упоминание о великом чешском композиторе. Работает это все на изящном сервисе D-BUS, который индексирует содержание документов и отслеживает изменения.

Также в разделе **Рабочий стол** вы найдете *PeaZip*, замечательную утилиту архивирования файлов, умеющую работать с атомным количеством форматов – каким бы ни был формат полученных от кого-то или скачанных из Интернет сжатых файлов, есть все шансы, что *PeaZip* справится с ним. Гуманные разработчики создали общий бинарный пакет, работать с которым на редкость просто: скопируйте файл **PeaZip1.4_Linux_x86_GTK2.tar.gz** в домашнюю директорию, затем откройте окно терминала и введите

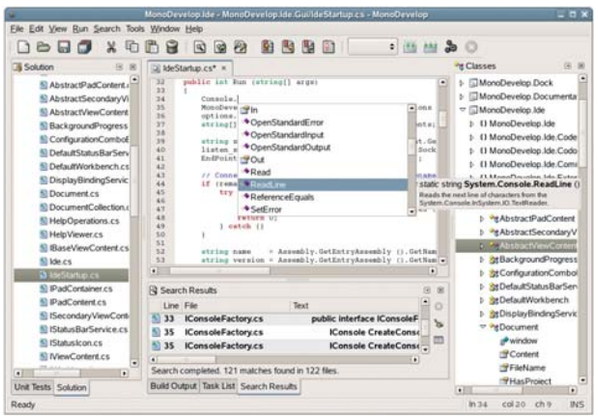
```
tar xfvz PeaZip1.4_Linux_x86_GTK2.tar.gz
PeaZip1.4_Linux_x86_GTK2/peazip
```

Этот скрипт распакует бинарный пакет и выполнит внутреннюю программу. Как только появится главное окно, можете создавать новый архив или открывать уже существующий.

Всякая всячина

В разделе **Интернет** вас ждёт встреча с *GPodder*, «медиа-агрегатором». Если вы не любитель умных слов Web 2.0, вам, возможно, захочется завизжать от такого описания *GPodder*. Не надо! Оно просто означает, что программа находит подкасты – медиа-файлы, разбросанные по всему Интернету. Например, LugRadio (www.lugradio.org) создает подкасты интервью с известными в среде Linux людьми, и вы можете насобирать их с помощью *GPodder*. На www.thelinuxlink.net вы найдете список относящихся к Linux подкастов, от руководств для новичков в Linux до обзоров свежих коммерческих игр.

Если вы – пользователь электронной почты IMAP и не слишком ладите со своим почтовым клиентом, познакомьтесь с *Mailody*. Вместо реализации всех мыслимых и немыслимых функций ее разработчики постарались создать быструю и аккуратную программу с функциональностью, нужной большинству пользователей.

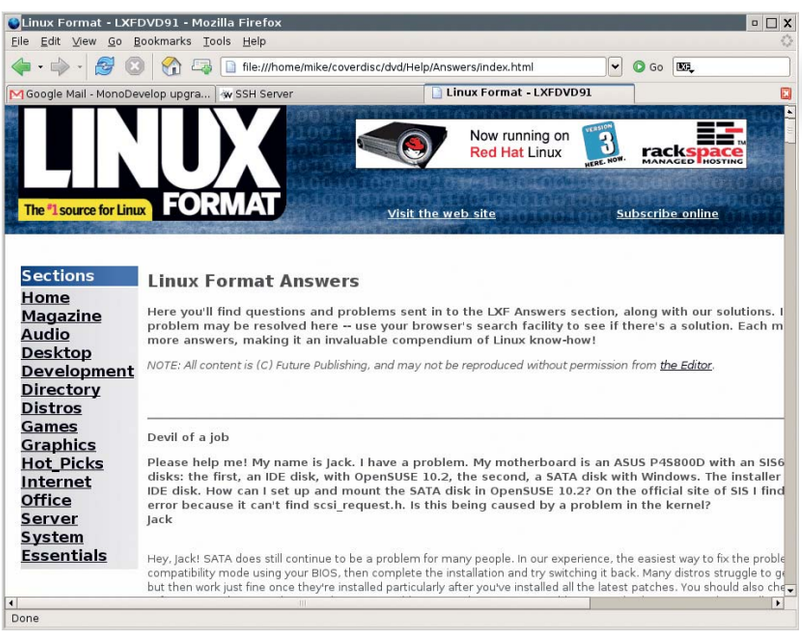


› ***MonoDevelop* приближается скачками, и имеет куда больше функций, чем сулит номер версии 0.13!**

В настоящее время она имеет функции маркировки и просмотра сообщений во вкладках. Исходный код релиза 0.41 – в разделе **Интернет/Mailody**. Для компиляции извлеките файл **Mailody-0.4.1.tar.bz2** и введите **./configure**, **make** и **make install** (как администратор) в появившейся директории. Поскольку это приложение KDE, перед компиляцией убедитесь, что у вас установлены библиотеки разработки KDE и *Qt*!

И, наконец, бегло упомянем о других жемчужинах DVD: если вы – поклонник KOffice, поупражняйтесь с исходным кодом 1.6.2 (раздел **Office**), он исправляет кучу ошибок и добавляет пару новых функций для *Krita* и *Kexi*. В разделе **Help** предлагается второй том нашего сборника Ответов** – все увеличивающийся список предлагаемых решений для проблем Linux. Если у вас возникли какие-то сложности, сначала загляните на диск, в Ответы – может быть, ваша проблема уже решена!

** На английском языке.



› **Проблемы с Linux? И некому помочь? Зайдите в раздел Help/Ответы, там решено более 70 типичных проблем Linux.**

И наконец...

Хочется чего-нибудь легонького? В разделе **Игры** вас ждет тщательно подготовленная подборка новых развлекательных релизов. Если вы – фанат красивых игр, вас порадует, что у нас есть имитатор бильярда… шутка. *Bygfoot* – футбольный менеджер, который позволяет вам покупать и продавать игроков, тренировать их, получать кредиты, использовать свой стадион и заниматься прочими менеджерскими штучками (например, встречаться в вестибюле отеля с корреспондентом желтой прессы, переодетым в шейха). Также предлагаем вам *Open Transport Tycoon Deluxe*, клон классической игры от Microprose, очень точно копирующий оригинал и имеющий несколько новых функций. Если вас интересуют стратегии, стоит взглянуть – вы найдете исходный код и пакеты Ubuntu в разделе **Игры/OpenTTD**. LXF

› **Не поддавайтесь опасному буму приватизации в Великобритании – покажите им, как это делается с помощью *OpenTTD*.**

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ DVD-ДИСК !

Информация о диске

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Содержание LINUX FORMAT DVD Май 2007

Сторона 1

Аудио
GSSMP – аудиоплеер для среды Gnome
Mixxx – цифровая диджейская система
Rosegarden – профессиональный секвенсор

Рабочий стол
CdFly – каталогизатор компакт-дисков
Kraft – программа для создания бизнес-документов
PeaZip – кроссплатформенный архиватор
Pinot – поисковая система для рабочего стола
Tasks – утилита для управления списком дел
TomBoy – липкие заметки
Veejay – видео-сэмплер

Разработка
GCC – коллекция компиляторов
Komodo – среда разработки для веб
Mono – свободная реализация .NET
MonoDevelop – среда разработки для Mono
Vala – новый язык для разработчиков Gnome-приложений

Дистрибутивы
Elive – LiveCD на базе Debian с рабочим столом Enlightenment
FreeBSD – популярная Unix-подобная ОС
FreeDOS – открытая DOS-совместимая ОС
FreeVMS – свободная реализация VMS
Haiku – клон BeOS
MikeOS – учебная ОС, написанная на ассемблере
Minix3 – ОС от Эндрю Танненбаума
NetBSD – Unix-подобная ОС с прозрачным дизайном
OpenBSD – BSD-система с упором в безопасность
ReactOS – клон Windows NT
Syllable – развитие AtheOS
Visopsys – альтернативная ОС, написанная с нуля

Игры
Brikx – головоломка
Bygfoot – футбольный менеджер
Cavez of Phear – клон «Диггера»
OpenTTD – клон Transport Tycoon Deluxe
Web of Life – стратегическая игра

Графика
Blender – профессиональный 3D-редактор
Fotox – программа для обработки изображений
Inkscape – векторный редактор
Simple Image Resizer – утилита для изменения размера изображений

Интернет
FlowPlayer – видеоплеер для Flash Video
gPodder – агрегатор подкастов
Ksniffer – сетевой сниффер
IjKlient – клиент для «Живого Журнала»
Mailody – почтовый клиент для IMAP
SmamAssassin – спам-фильтр

Офис
Koffice – офисный пакет для KDE
OpenOffice.org – сборка OpenOffice.org от компании «Инфра-Ресурс»
PDFEdit – редактор pdf
Scribus – открытая программа для верстки

Сервер
Ant – система сборки на Java
Atutor – система управления обучением
Rabbit – кэширующий прокси-сервер
SugarCRM – открытая версия SugarCRM
Webalizer – анализатор логов веб-сервера

Система
Gnome Sensors Applet – апплет для отображения информации lm_sensors
Ksensors – интерфейс к lm_sensors для KDE
lm_sensors – пакет для получения данных с аппаратных датчиков
QEMU – свободный эмулятор процессора
VirtualBox – свободая версия виртуальной машины

Поставляется вместе с журналом LINUXFORMAT номер 5(92) Май 2007

Сторона 2

Arch Linux 0.8

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен на Уральском электронном заводе., 620066, Россия, г. Екатеринбург, ул. Комвузовская 17-203, Лицензия МПТР России ВАФ № 77-15

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

Содержание DVD

ЖУРНАЛ
GTK ..... Код примеров статьи.
JavaEE ..... Код примеров статьи.
Mono ..... Код примеров статьи.
PDFs ..... Статьи предыдущих номеров LXF.
Roundup ..... Web-редакторы.
SugarCRM ..... Код примеров статьи.
Unix API ..... Код примеров статьи.

ЗВУК
GSSMP ..... Музыкальный плейер Gnome.
Mixxx ..... Цифровой инструмент для DJ.
Rosegarden ..... MIDI-секвенсер.

РАБОЧИЙ СТОЛ
CdFly ..... Менеджер коллекции CD.
Kraft ..... Программа управления малым бизнесом.
PeaZip ..... Инструмент сжатия.
Pinot ..... Инструмент поиска для рабочего стола.
Tasks ..... Список планируемых дел.
Tomboy ..... Блокнот для заметок.
Veejay ..... Видеосэмплер.

РАЗРАБОТКА
GCC ..... Компилятор GNU.
Komodo IDE ..... Пакет Web разработки (21-дневная пробная версия).
Mono ..... Среда .NET с открытым кодом.
MonoDevelop ..... Среда интегрированной разработки C#.
Vala ..... Компилятор системы типов Gobject.

ДИСТРИБУТИВЫ
Elive ..... Дистрибутив Live на базе Enlightenment.
FreeBSD ..... ОС для рабочих столов и серверов.
FreeDOS ..... DOS-совместимая ОС.
FreeVMS ..... Клон VMS с открытым кодом.
Haiku ..... Клон BeOS с открытым кодом.
MikeOS ..... Написанная на Ассемблере ОС реального режима.
Minix3 ..... Легковесная Unix-подобная ОС.
NetBSD ..... Ультрапереносимая ОС.
OpenBSD ..... Супербезопасная ОС.
ReactOS ..... Windows-совместимая ОС.
Syllable ..... Настольная операционная система.
Visopsys ..... Графическая мини-ОС.

ИГРЫ
Brikx ..... Головоломка.
Bygfoot ..... Футбольный менеджер.
Cavez of Phear ..... Клон Boulderdash в командной строке.

› Visopsys умудряется впихнуть графическую ОС на одну-единственную дискету!

OpenTTD ..... Стратегия, вдохновленная Transport Tycoon.
Web of Life ..... Симулятор Жизни.

ГРАФИКА
Fotox ..... Программа модификации изображений.
Inkscape ..... Редактор векторной графики.
Simple Image Resizer ..... Утилита масштабирования изображений.

СПРАВКА
Answers ..... Решенные проблемы Linux.
Rute ..... Руководство по администрированию Linux.

HOTPICKS
AcetoneISO ..... Манипулятор образов диска.
ClamAV ..... Интерфейс антивируса.
GREYCstoration ..... Улучшатель изображений.
KDESvn ..... Интерфейс пользователя Subversion.
KeyFrog ..... Монитор использования клавиатуры.
Nagios ..... Сетевой монитор.
Rawstudio ..... Манипулятор файлов в формате RAW.
TuxWordSmith ..... Scrabble-подобная игра.
UFO Alien Invasion ..... Стратегия.
XML Copy Editor ..... Редактор XML.

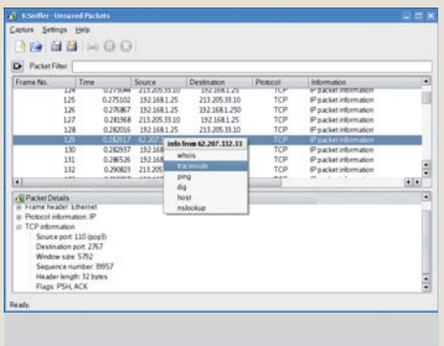
ИНТЕРНЕТ
FlowPlayer ..... Проигрыватель FLV Flash.
GPodder ..... Поиск подкастов.
KSniffer ..... Сетевой сниффер.
LJKlient ..... Клиент KDE LiveJournal.
Mailody ..... Почтовый клиент на основе Qt/KDE.
SpamAssassin ..... Спам-фильтр с открытым кодом.

ОФИС
KOffice ..... Офисный пакет KDE.
PDFEdit ..... Редактор PDF.
Scribus ..... Настольная издательская система.

СЕРВЕР
Ant ..... Инструмент разработки на основе Java.
Rabbit ..... Web прокси-сервер.
SugarCRM ..... CRM-система.
Webalizer ..... Анализатор журналов web-сервера.

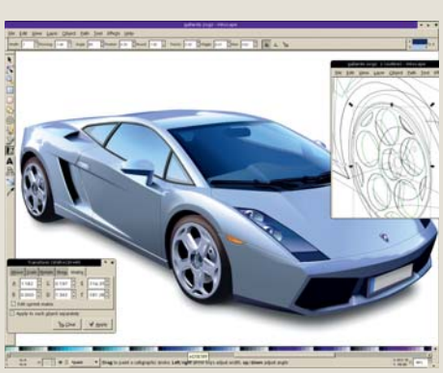
СИСТЕМА
Gnome Sensors Applet Сбор информации об оборудовании.
KSensors ..... Интерфейс пользователя Lm-sensors в KDE.
LibFerris ..... Виртуальная файловая система.
Lm-sensors ..... Инструмент мониторинга оборудования.
Qemu ..... Эмулятор ПК.
Simple Linux Backup ... Утилита создания резервных копий и восстановления.
VirtualBox ..... Инструмент виртуализации и эмулятор ПК.

ГЛАВНОЕ
ATI driver ..... Графический драйвер.
Avifile ..... Библиотека чтения/записи AVI-файлов.
Bash ..... Командный интерпретатор.
CheckInstall ..... Программа создания двоичных пакетов.



› Следите за своей сетью – *KSniffer* отловит злонамеренных пользователей.

Coreutils ..... Утилиты командной строки.
CSV ..... Содержание диска.
GLib ..... Низкоуровневая библиотека C.
Glibc ..... Библиотека GNU C.
GTK ..... Инструментарий пользовательского интерфейса.
HardInfo ..... Системная информация и сравнительный анализ.
Jigdo ..... Создатель образов ISO.
Kernel ..... Свежий релиз ядра Linux.
LibXML2 ..... XML-анализатор и набор инструментов.
Ncurses ..... Оконный инструментарий текстового режима.
Nvidia driver ..... Графический драйвер.
Python ..... Язык программирования.
Rawrite ..... Программа записи образов дискет.
SBM ..... The Smart Boot Manager.
SDL ..... Библиотека мультимедиа.



› *Inkscape 0.45* усовершенствовал производительность и обзавелся Историей.

Школа LINUX FORMAT

Мы открываем новый раздел журнала, посвященный использованию Linux и прочих продуктов с открытым кодом в сфере образования. В последнее время эта проблема становится все более и более актуальной, вопрос встает ребром. Linux Format готов внести свой посильный вклад в переход российского образования на свободные рельсы.

Материалы раздела будут отражать текущую ситуацию в части продвижения Linux и open source в учебный процесс образовательных учреждений всех уровней. Мы следим за возникающими проблемами, запросами, требованиями и решениями в этом стратегически важном деле. И, конечно – за людьми, которые своими руками и своим интеллектом продвигают свободное программное обеспечение в нашу повседневную жизнь.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Open source- продукты и технологии для преподавателя

Пожарина Галина Юрьевна,
начальник отдела информационных технологий,
Санкт-Петербургский технический колледж управления и коммерции

Любое образование немыслимо без так называемых информационных технологий. И не потому, что это модно и современно. ИТ настолько внедрились в нашу жизнь, что исключи их хоть на день, хоть на час – мы испытываем дискомфорт. В офисе «упал Office» - парализована работа, перебои связи в ICQ – менеджеры в простое, засбоил принтер – бухгалтерия снижает темпы. Знакомая картина? Самая уважительная для преподавателей причина несдачи вовремя курсовиков и дипломных работ – неполадки с компьютером, с Интернетом у студентов. Поэтому ИТ и сопровождают образование с самых азов до высоко профессионально-ориентированных направлений.

Подбор программного обеспечения для образовательного процесса – непростой и неординарный процесс. Большая часть создаваемого программного обеспечения пригодна для обучения, для использования в школах, колледжах, вузах. Это совершенно не обязательно «электронные учебники» и энциклопедии, тесты и анкеты. Ориентироваться в библиотеках ПО достаточно сложно, тем более что картина постоянно меняется – ведь авторы программных продуктов не сидят сложа руки.

На подбор программных сред и комплексов для реализации различных составляющих любого образования влияют следующие факторы:

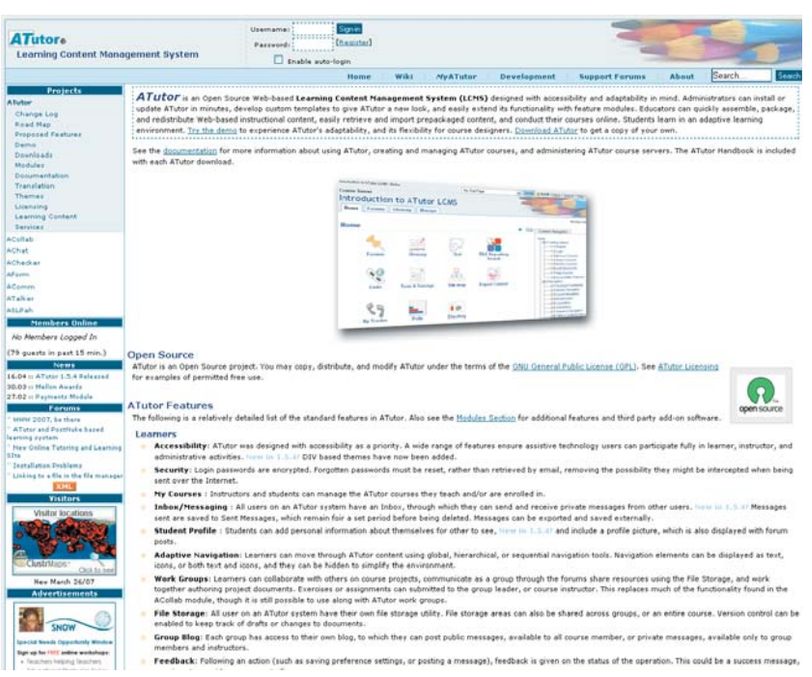
- **Правовой** (лицензионное право на использование, а также распространение программных продуктов, в том числе и для организации самостоятельной работы студентов);
- **Экономический** (стоимость лицензионного программного продукта, его обновлений, стоимость его технического сопровождения и пр.);
- **Функциональный** (перечень потенциально возможных операций, их соответствие функциональным обязанностям сотрудников, перечню умений студентов, определенных государственным образовательным стандартом);
- **Эргономический** (удобность для пользователя, в том числе преемственность версий продукта);
- **Фактор стабильности** (устойчивость к вирусным атакам, обеспечение сохранности данных при неполадках техники и при несанкционированном доступе к данным).

Open Source ПО позволяет подготовить специалиста, который не ограничен в выборе конкретной реализации, он способен провести анализ и выбрать наиболее подходящий программный продукт для решения поставленный перед ним задачи.

Анализ рынка Open Source ПО и рекомендации по интеграции его в обучение, как мы рассчитываем, станет полезным как для преподавателей, так и для руководителей образовательных учреждений.

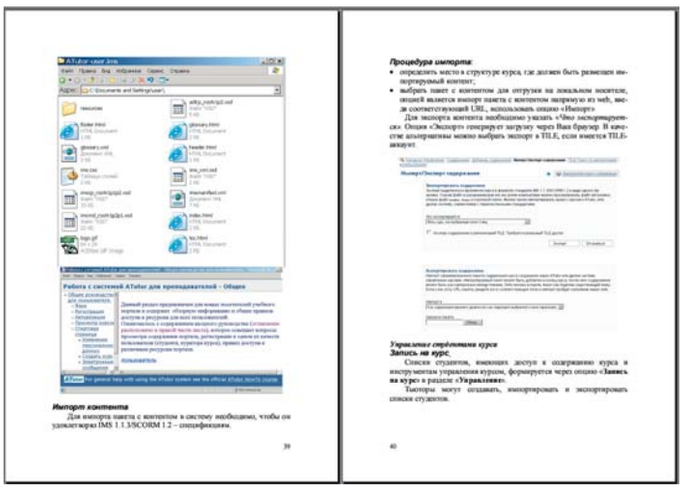
Организация дистанционного обучения на базе современных информационных и коммуникационных технологий становится в современных условиях важной составляющей процесса информатизации образовательного учреждения. Управление динамическим образовательным контентом может быть решено на базе как коммерческих продуктов, таких как DLS, e-University, REDCLASS, Stellus, WebTutor, СДО «Прометей» и др., так и продуктов, распространяемых на основе генеральной лицензии GPL – *ATutor*, *Caroline*, *LogiCampus*, *Moodle*, *Segue*, *Site@School* или пр.

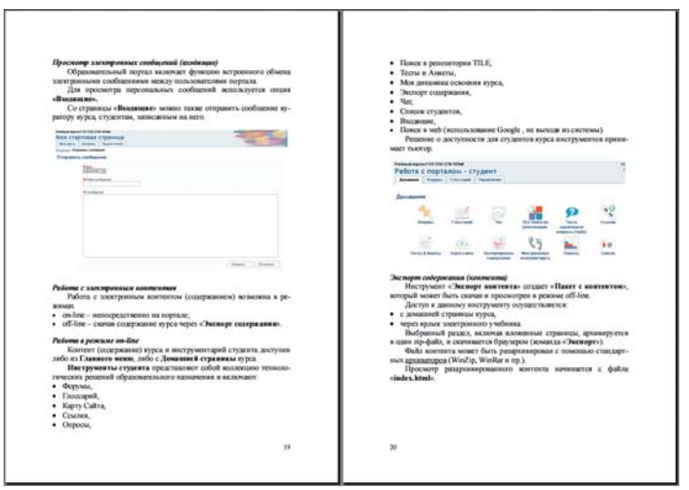
Система управления образовательным контентом *ATutor* (сайт разработчика http://www.atutor.ca) отличается не только интеграцией с системами управления контентом *PostNuke* и *Mambo*, но и русской локализацией (стабильная версия *ATutor1.5.2*), что дает право рекомендовать ее для широкого использования. *ATutor* базируется на Web Системе Управления Контентом Обучения (LCMS) разработанной с возможностью обеспечения доступа и адаптации по желанию преподавателя. Администратор может проинсталлировать или обновить *ATutor* за минуты. Преподаватели могут быстро собрать, упаковать, и перенести учебные курсы, и открыть их в сети. Студенты получают адаптируемую среду обучения. Помимо online использования электронных курсов, система позволяет выгружать в offline отдельные разделы, либо целые курсы по усмотрению преподавателя – тьютора, что позволяет и расширить аудиторию учащихся, и разнообразить способы применения обучающего контента. К преимуществам системы относится также и оптимальный набор инструментов для преподавателя, позволяющий не только вести мониторинг активности учащихся, но и самостоятельно определять степень доступа к курсу для разных категорий пользователей.

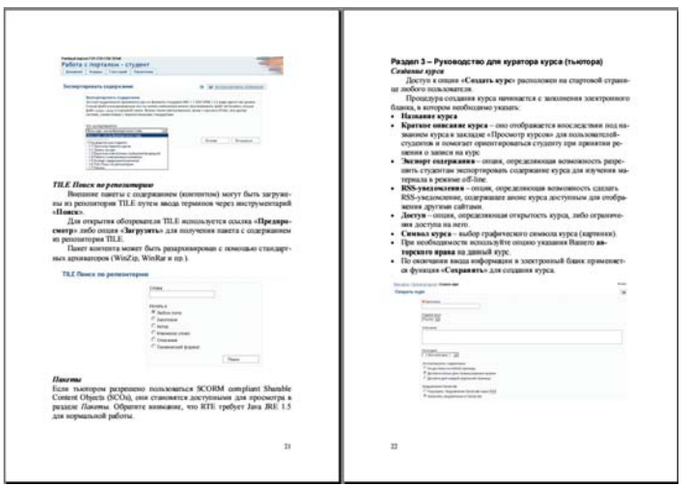


Методички LXF

Дистанционное обучение на базе системы ATutor







С этого номера мы начинаем публиковать методические материалы Г.Ю.Пожариной.

Это пособие вы найдете на нашем диске в файле **education.pdf.**

Продолжение – в следующих номерах.

Линуксцентр представляет авторизованные курсы Mandriva Linux.

Курсы не привязаны к конкретному дистрибутиву и сертифицированы международной организацией LPI - Linux Professional Institute (www.lpi.org).

★ LNX70: Введение в Linux.

Предназначен для пользователей, которые хотят приобрести навыки использования Linux. В рамках курса рассматриваются вопросы миграции на Linux и даются базовые знания для пользователей.

Цель курса: Обучить пользователя оптимальной работе с Linux.
Требования к соискателям: Базовая компьютерная подготовка.

★ LPI101: Администрирование Linux для начинающих.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования Linux. Рассматриваются следующие вопросы: конфигурирование системы, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

Цель курса: Подготовить системного администратора Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI101.

Требования к соискателям: Знания по Linux в объеме курса LNX70.

★ LPI102: Сетевое администрирование Linux.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования сетей на базе Linux. Рассматриваются следующие вопросы: конфигурирование периферийных устройств, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

Цель курса: Подготовить системного администратора к управлению сетями на базе Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI102.

Требования к соискателям: Знания по Linux в объеме курса LPI101.

★ LPI201: Администрирование Linux для специалистов.

Курс предназначен для системных администраторов, которые хотят повысить свою квалификацию в области управления сложными Linux-системами. Рассматриваются следующие вопросы: настройка ядра Linux, восстановление системы, восстановление файловой системы, настройка RAID, сопровождение системы, автоматизация задач администрирования.

Цель курса: Подготовить системного администратора к управлению сложными Linux-системами в соответствии с требованиями к соискателям на сдачу экзамена LPI201.

Требования к соискателям: Знания по Linux в объеме курса LPI101 или LPI102.

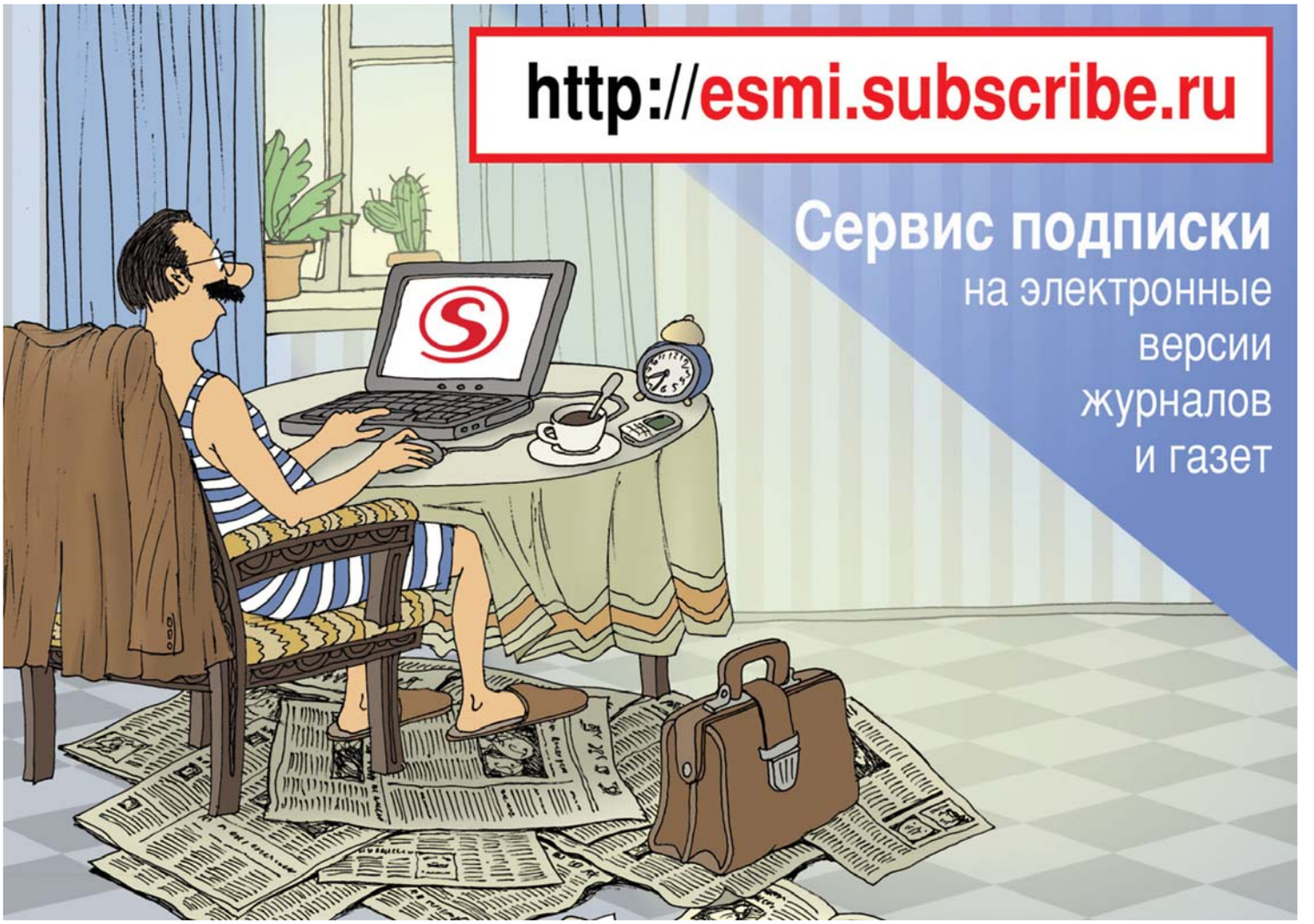
★ LPI202: Сетевое администрирование Linux для специалистов.

Курс предназначен для опытных системных администраторов, которые хотят усовершенствовать навыки управлениями сетевыми сервисами на базе Linux. Рассматриваются следующие вопросы: углубленное конфигурирование сети, настройка служб электронной почты и новостей, расширенная настройка DNS, обслуживание интернет-сервера, настройка DHCP, NIS, LDAP и PAM, управление безопасностью сетевых служб, контроль за трафиком.

Цель курса: Подготовить системного администратора к управлению сложными сетевыми сервисами в соответствии с требованиями к соискателям на сдачу экзамена LPI202.

Требования к соискателям: Знания по Linux в объеме курса LPI101.

Узнайте подробности и закажите курсы на сайтах - www.linuxcenter.ru и www.unixedu.ru



В «Системном администраторе» вы не прочтете о:

– котировках валют
– сплетнях
– погоде
– политике
– развлечениях

В вашем распоряжении:

– опыт лучших IT-специалистов
– новые идеи и полезные советы
– самые эффективные решения в области системного и сетевого администрирования

Подпишитесь сейчас!

Роспечать – 20780, 81655
Пресса России – 87836
Online-подписка – www.linuxcenter.ru

Время подписки ограничено!

LINUX FORMAT
Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Родион Водейко, Иван Мищенко, Елена Толстякова
Переводчики
Александр Бикмеев, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Валентин Развозжаев, Сергей Супрунов, Александр Черных, Юлия Шабунио
Допечатная подготовка
Мария Пучкова, Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____
Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Литературный редактор
Ребекка Смолли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Штатный автор
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Ассистент по выпуску
Эндрю Грегори (Andrew Gregory) andrew.gregory@futurenet.co.uk

Авторы
Марк Бейн (Mark Bain), Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Д-р Крис Браун (Dr Chris Brown), Энди Ченнел (Andy Channelle), Дэвид Кулсон (David Coulson), Скотт Дуглас (Scott Douglass), Энди Хадсон (Andy Hudson), Бен Мартин (Ben Martin), Евгений Балдин, Андрей Боровский, Иван Зорин, Андрей Прахов, Алексей Федорчук, Антон Черноусов, Илья Шпаньков
Художественные ассистенты: Род Хейл (Rob Hale), Сет Син (Seth Singh)
Фотографии: Джена Камбо (Jena Cumbo)
Иллюстрации: Нейл Бартлет (Neil Bartlett), Пол Блехфорд (Paul Blachford), Крис Винн (Chris Winn)

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция): ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Представительство в Москве:
пр.Мира, 161, телефон +7(495) 799-18-63, +7(495)136-88-45
Email: info@linuxformat.ru, Web: www.linuxformat.ru

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензией Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

За содержание рекламных материалов редакция ответственности не несет.

Linux-зарегистрированная торговая марка Линуса Торвальдса (Linus Torvalds). Название «GNU/Linux» заменяется на «Linux» в целях сокращения. Остальные торговые марки являются собственностью их законных владельцев.
Linux Format является торговой маркой Future Publishing Ltd (Future plc group company).

За информацией о журналах, издаваемых Future plc group company, обращайтесь
http://www.futureplc.com

© Linux Format 2005
© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

САМ СЕБЕ СИСАДМИН!

Присматриваете ли вы за парой ПК, или за парой тысяч – мы предлагаем вам советы по повышению производительности, безопасности и сетям!

Kamion, упрощающий миграцию

Читайте наш обзор инструмента для переноса пользовательских данных, который готовится к включению в KDE4.

И все это о SELinux

Щит, созданный Национальным Агентством Безопасности США, теперь доступен каждому.

Джереми Эллисон

О разработке Samba, новой жизни в Google и причинах ухода из Novell вскоре после подписания соглашения с Microsoft.

Содержание последующих выпусков может меняться без уведомления

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА В ЛИНУКСЦЕНТРЕ

Сколько стоит подписка?

Подписка на журнал «Linux Format» 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь) стоит 1800 рублей
Подписка на журнал «Linux Format» 6 номеров (июль, август, сентябрь, октябрь, ноябрь, декабрь 2006 года) стоит 900 рублей

Как оформить подписку?

Чтобы оформить подписку на журнал «Linux Format», необходимо зарегистрироваться в интернет-магазине Linuxcenter.Ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар «Подписка на журнал «Linux Format» 12 номеров 2006 года», или товар «Подписка на журнал «Linux Format» второе полугодие 2006 года», получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Как оплатить подписку?

– по выставленному счету (для юридических лиц)
– по квитанции в любом отделении Сбербанка

Плюсы подписки

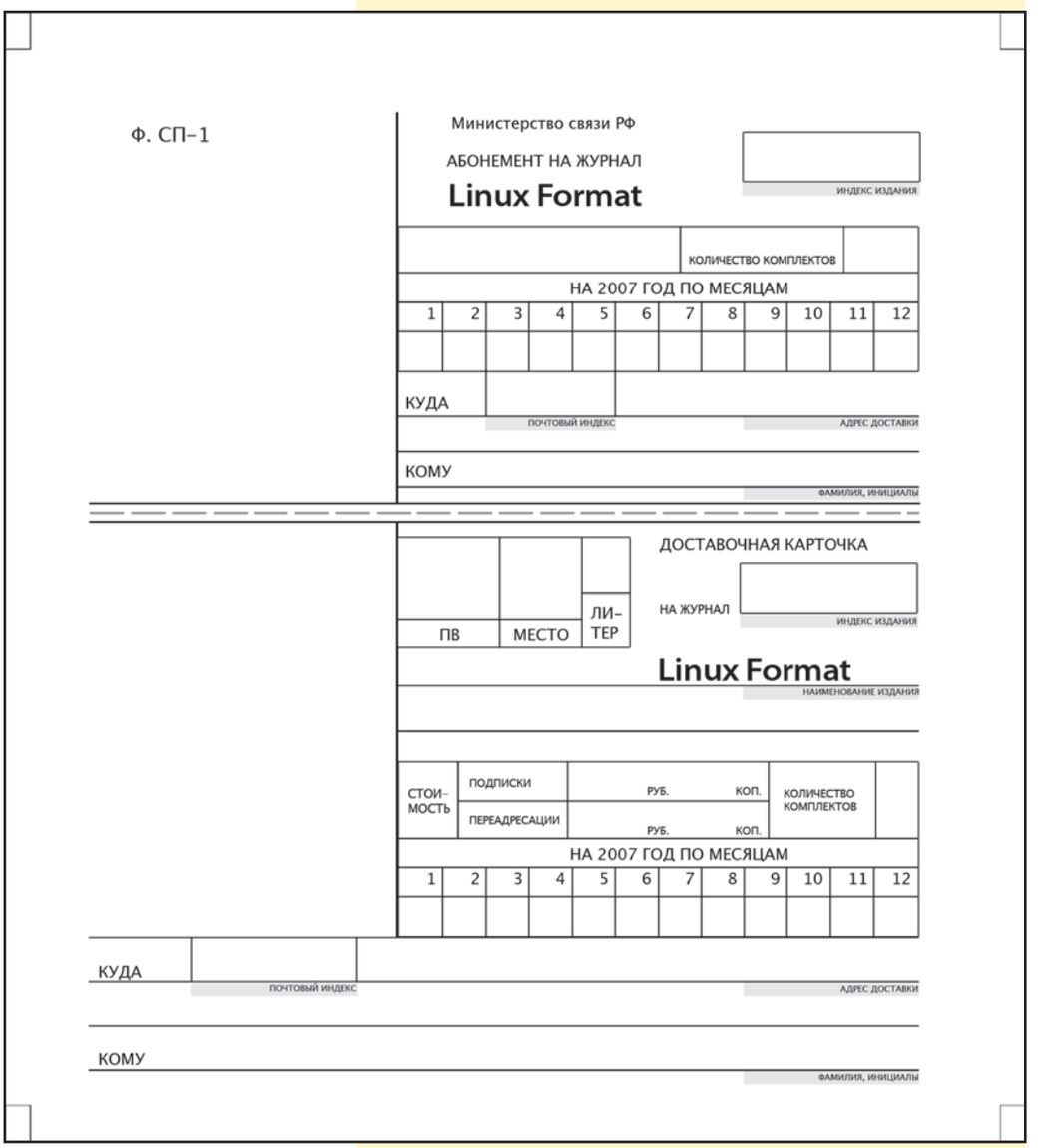
- подписка дешевле!
- гарантированное получение нового номера журнала!

ПОДПИСКА - 2007!

ПОДПИСКА ПО КАТАЛОГАМ РФ

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882

Каталог «ПРЕССА РОССИИ» – подписной индекс 87974



Ф. СП-1

Министерство связи РФ

АБОНЕМЕНТ НА ЖУРНАЛ

Linux Format

ИНДЕКС ИЗДАНИЯ

КОЛИЧЕСТВО КОМПЛЕКТОВ

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА

ПОЧТОВЫЙ ИНДЕКС

АДРЕС ДОСТАВКИ

КОМУ

ФАМИЛИЯ, ИНИЦИАЛЫ

ДОСТАВОЧНАЯ КАРТОЧКА

НА ЖУРНАЛ

ИНДЕКС ИЗДАНИЯ

ПВ | МЕСТО | ЛИТЕР

Linux Format

НАИМЕНОВАНИЕ ИЗДАНИЯ

СТОИМОСТЬ | ПОДПИСКИ | РУБ. КОП. | КОЛИЧЕСТВО КОМПЛЕКТОВ

ПЕРЕАДРЕСАЦИИ | РУБ. КОП.

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА

ПОЧТОВЫЙ ИНДЕКС

АДРЕС ДОСТАВКИ

КОМУ

ФАМИЛИЯ, ИНИЦИАЛЫ

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА ПО КАТАЛОГАМ СНГ И БЛИЖНЕГО ЗАРУБЕЖЬЯ

Каталог «Российская Пресса» – совместный проект Государственного предприятия «Казпочта» , Агентства «Книга-Сервис» и АРЗИ.

Блок изданий АРЗИ в национальных Каталогах Украины и Беларуси. В Азербайджане, Армении, Грузии, Киргизии, Узбекистане и Молдове - по изданиям, включенным в Объединенный каталог, распространяемые через АРЗИ.

Азербайджан
- по Объединенному каталогу российских изданий через Предприятие по распространению печати «Гасид»(370102, г. Баку, ул. Джавадхана, 21);

Армения
- по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г.Ереван, пл.Сасунци Давида, д.2) и ЗАО «Контакт-Мамул» (375002, Г.Ереван, ул.Сарьяна, 22);

Белоруссия
- по Каталогу изданий стран СНГ через РГО «Белпочта» (220050, г.Минск, пр-т Ф.Скорины, 10);

Грузия
- по списку номенклатуры «АРЗИ»» через АО «Сакпресса» (380019, г.Тбилиси, ул.Хошараульская, 29) и АО «Мацне» (380060, г.Тбилиси, пр-т Гамсахурдия, 42);

Казахстан
- по Каталогу «Российская Пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»;

Молдавия
- по каталогу через ГП «Пошта Молдавей» (МД-2012, г.Кишинев, бул.Штефан чел Маре, 134);
- по списку через ГУП «Почта Приднестровья» (MD-3300, г.Тирасполь, ул.Ленина, 17);
- по прайс-листу через ООО Агентство «Editil Periodice» (2012, г.Кишинев, бул. Штефан чел Маре, 134).

Узбекистан
- по Каталогу «Davriy nashrlar» российские издания через Агентство по распространению печати «Davriy nashrlar» (7000029, Ташкент, пл.Мустакиллик, 5/3, офис 33);

Украина
- Киевский главпочтамт.
- Подписное агентство «KSS» Телефон/факс (044)270-62-20, 270-62-22

ПОДПИСКА НА LINUX FORMAT

Агентство "Centerpress"

Сколько стоит подписка?
Подписка на журнал "Linux Format" 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь 2007 года) стоит 1800 рублей.

Как оформить подписку?
Чтобы оформить подписку на журнал "Linux Format", необходимо зарегистрироваться в интернет-агентстве Centerpress.ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар "Подписка на журнал "Linux Format" на 2007 год 12 номеров (01-12 / 2007)", получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Агентство "Centerpress" : www.centerpress.ru

Все Плюсы подписки!
- Подписка дешевле!
- Гарантированное получение журнала!

По каталогам РФ

Каталог агентства "РОСПЕЧАТЬ" - подписной индекс

20882

Каталог "ПРЕССА РОССИИ" - подписной индекс

87974

АЛЬТЕРНАТИВНЫЕ АГЕНТСТВА РФ

Агентство «Интер-Почта»
(095) 500-00-60, курьерская доставка по Москве.

Агентство «Вся Пресса»
(095) 787-34-47

Агентство «УралПресс»

- **Екатеринбург, Березовский, В. Пышма, Первоуральск тел. (343) 375-80-71, 375-84-93, 375-84-39, факс 375-62-74, info@ural-press.ru**
- **Нижний Тагил тел. (3435) 411448, 417709, ntagil@ural-press.ru**
- **Челябинск тел. (351) 262-90-03, 262-90-05, pochta@chel.surnet.ru**
- **Пермь тел. (3422) 60-24-40, 60-22-95, 60-35-42, parma-press@permonline.ru**